

Interrupt List, part 2 of 18

Copyright (c) 1989-1999,2000 Ralf Brown

-----V-10BF--CX0000-----

INT 10 - Athena Digital HGCIBM.COM - INSTALLATION CHECK

AH = BFh

CX = 0000h (???)

Return: CX = 0202h

DL = ??? (internal data, possibly version number)

-----b-10BF00-----

INT 10 - VIDEO - Compaq Extensions - SELECT EXTERNAL MONITOR

AX = BF00h

Return: nothing

Desc: specify that the external monitor become the active monitor

Note: all registers preserved and the internal monitor is blanked

SeeAlso: AX=BF01h,AX=BF12h,AH=12h/BL=35h

-----b-10BF01-----

INT 10 - VIDEO - Compaq Extensions - SELECT INTERNAL MONITOR

AX = BF01h

Return: nothing

Desc: specify that the internal monitor become the active monitor

Note: all registers preserved and the external monitor is blanked

SeeAlso: AX=BF00h,AX=BF12h,AH=12h/BL=35h

-----b-10BF02-----

INT 10 - VIDEO - Compaq Extensions - SET MASTER MODE OF CURRENT CONTROLLER

AX = BF02h

BH = master mode

04h CGA

05h EGA

07h MDA

Return: nothing

SeeAlso: AX=BF03h

-----b-10BF03BX0000-----

INT 10 - VIDEO - Compaq Extensions - GET ENVIRONMENT

AX = BF03h

BX = 0000h

Return: BH = active monitor

(00h = external, 01h = internal, 03h both, 04h neither)

BL = master mode (see #00211)

CH = ASIC type and version

00h (reserved, very early Compaq VGCs)

31h if QVision VGA

CL = switchable VDU modes supported (see #00212)

DH = internal monitor type (see #00213)

DL = external monitor type (see #00213)

SeeAlso: AX=1A00h,AX=BF00h,AX=BF01h,AX=BF02h,AX=BF11h

(Table 00211)

Values for Compaq video master mode:

00h switchable VDU not present

04h CGA

05h EGA

07h MDA

08h switchable LCD controller present

09h plasma VGA

0Ah TFT (active-matrix) VGA

Bitfields for Compaq switchable VDU modes support:

Bit(s) Description (Table 00212)

0 CGA supported

1,2 reserved (1)

3 MDA supported

4 BitBLT engine available

5 132-column support available

6 640x480x256 mode available

7 8-bit DAC mode available

(Table 00213)

Values for Compaq monitor type:

00h none

01h dual-mode monitor

02h 5153 RGB monitor (RGBI 16-color)

03h Compaq Color monitor

04h 640x400 flat panel (Plasma CGA)

05h VGC monochrome

06h VGC color

07h 8-level mono LCD VGA (internal)

1024x768 mono VGA (external)

08h 16-level mono plasma VGA (internal)

1024x768 color VGA (external)

09h 4-level mono LCD CGA

0Ah 16-level mono LCD VGA

0Bh active-matrix color VGA

0Ch active-matrix mono VGA

0Dh STN color VGA

-----b-10BF04-----

INT 10 - VIDEO - Compaq Extensions - SET MODE SWITCH DELAY

AX = BF04h

BH = new state of delay (00h enabled, 01h disabled)

Return: nothing

Note: the 1993/04/08 sytem ROM appears to use BL instead of BH; a future
version of the list will provide more definite information

SeeAlso: AX=BF05h

-----b-10BF05-----

INT 10 - VIDEO - Compaq Extensions - ENABLE/DISABLE DISPLAY

AX = BF05h

BH = new state of video

00h off

01h on

Return: nothing

Note: the 1993/04/08 sytem ROM appears to use BL instead of BH; a future
version of the list will provide more definite information

SeeAlso: AH=12h/BL=36h,AX=BF04h

-----b-10BF06-----

INT 10 - VIDEO - Compaq SLT/286 - READ GRAY SCALE TABLE

AX = BF06h

CL = address to be read from gray scale table

Return: AL = bit 3-0 - Value read from gray scale table

CL = address to be read from gray scale table

SeeAlso: AH=12h/BL=33h,AX=BF07h

-----b-10BF07-----

INT 10 - VIDEO - Compaq SLT/286 - WRITE GRAY SCALE TABLE

AX = BF07h

CH = value to write to gray scale table

CL = address to be written to gray scale table

Return: nothing

SeeAlso: AX=BF06h,AX=BF08h

-----b-10BF08-----

INT 10 - VIDEO - Compaq SLT/286 - WRITE COLOR MIX REGISTERS

AX = BF08h

CH = bits 7-4 green weight

bits 3-0 blue weight

CL = bits 7-4 unused

bits 3-0 red weight

Return: nothing

SeeAlso: AH=12h/BL=33h,AX=BF07h

-----b-10BF09-----

INT 10 - VIDEO - Compaq Extensions - TURN ON REVERSE VIDEO

AX = BF09h

Return: CF clear

Notes: sets bit 6 of port 03CEh index 81h; in some cases also sets index 84h
to 6Fh and index 83h to 04h

supported by at least the QVision board and the LTE Lite ROM BIOS

SeeAlso: AX=BF0Ah,AX=BF0Bh

-----b-10BF0A-----

INT 10 - VIDEO - Compaq Extensions - TURN OFF REVERSE VIDEO

AX = BF0Ah

Return: CF clear

Notes: clears bit 6 of port 03CEh index 81h; in some cases also sets index 84h
to 80h and index 83h to FBh

supported by at least the QVision board and the LTE Lite ROM BIOS

SeeAlso: AX=BF09h,AX=BF0Bh

-----b-10BF0B-----

INT 10 - VIDEO - Compaq Extensions - CHECK WHETHER VIDEO REVERSED

AX = BF0Bh

Return: CF clear

AX = state (0000h normal video, 0001h reverse video)

Note: supported by at least the QVision board and the LTE Lite ROM BIOS

SeeAlso: AX=BF09h,AX=BF0Ah

-----b-10BF0C-----

INT 10 - VIDEO - Compaq Extensions - SET DAC 6-BIT MODE

AX = BF0Ch

Return: CF clear

Desc: specify that video DAC registers use only six bits of color info

Note: supported by at least the QVision board and the SystemPro XL ROM BIOS

SeeAlso: AX=BF0Dh,AX=BF0Eh

-----b-10BF0D-----

INT 10 - VIDEO - Compaq Extensions - SET DAC 8-BIT MODE

AX = BF0Dh

Return: CF clear

Desc: specify that video DAC registers use all eight bits of color info

Note: supported by at least the QVision board and the SystemPro XL ROM BIOS

SeeAlso: AX=BF0Ch,AX=BF0Eh

-----b-10BF0E-----

INT 10 - VIDEO - Compaq Extensions - CHECK DAC 6-BIT/8-BIT mode

AX = BF0Eh

Return: CF clear

AL = DAC mode

00h in 6-bit mode

01h in 8-bit mode

Note: supported by at least the QVision board and the SystemPro XL ROM BIOS

SeeAlso: AX=BF0Ch,AX=BF0Dh

-----b-10BF0F-----

INT 10 - VIDEO - Compaq Extensions - SET HIGH ADDRESS MAP REGISTER

AX = BF0Fh

BX = high address map location in megabytes, or

0000h to disable high address map, or

FFFFh to let ROM configure high address map

Return: CF clear

AX = previous value of high address map register

Notes: supported by at least the QVision board and the SystemPro XL and LTE

Lite ROM BIOSes

SeeAlso: AX=BF10h

-----b-10BF10-----

INT 10 - VIDEO - Compaq Extensions - GET HIGH ADDRESS MAP REGISTER

AX = BF10h

Return: CF clear

AX = current value of high address map register

Notes: supported by at least the QVision board and the SystemPro XL and LTE

Lite ROM BIOSes

the Compaq QVision documentation (October 1993) says the value is

returned in BX, but this appears to be one of many typos

SeeAlso: AX=BF0Fh

-----b-10BF11-----

INT 10 - VIDEO - Compaq Extensions - GET EXTENDED ENVIRONMENT

AX = BF11h

Return: AL = BFh if supported

CF clear

ES:DI -> ASCII signature "COMPAQ"

ES:SI -> advanced functionality table (see #00214)

Note: supported by at least the QVision board and the LTE Lite and ProSignia

ROM BIOSes

SeeAlso: AX=1A00h,AX=BF03h

Bitfields for Compaq advanced video functionality:

Bit(s) Description (Table 00214)

31 future graphics extensions (reserved, set to 0)
30-16 reserved
15-12 available video memory in 256K blocks, less one (0000 = 256K, etc.)
11-8 reserved
7 QVision modes supported
6 Advanced VGA modes supported
5 Accelerated VGA modes supported
4 standard VGA modes supported
3-2 reserved
1 132-column modes supported
0 reserved

-----b-10BF12-----

INT 10 - VIDEO - Compaq Extensions - NEW ACTIVE MONITOR

AX = BF12h

BH reserved

BL = function mask (see #00215)

Return: CF clear

Notes: this function is a NOP if the VGA subsystem is inactive or the current

Display Combination Code is 10h or higher

supported by at least the QVision board and the LTE Lite and ProSignia

ROM BIOSes

SeeAlso: AX=BF01h,AX=BF02h,#00733

Bitfields for function mask:

Bit(s) Description (Table 00215)

7 command mode

if set, bits 1 and 0 make the monitor active when set

if clear, bits 1 and 0 toggle the monitor's state when set

6-2 reserved

1 internal monitor

0 external monitor

-----b-10BF13-----

INT 10 - VIDEO - Compaq QVision - GAMMA CORRECTION

AX = BF13h

BL = subfunction

00h load gamma correction table for true-color mode

DS = BIOS data segment

01h set palette RAM bypass

BH = 00h

Return: nothing

Note: these functions must be called after a video mode set, and are in

effect only for the current video mode

SeeAlso: #00733

-----V-10BFA0BXADAD-----

INT 10 - VIDEO - Compaq ADAPT.COM - INSTALLATION CHECK

AX = BFA0h

BX = ADADh

Return: AX = BDBDh if newer ADAPT.COM installed

BX = BCD version (BH = major, BL = minor)

CL = ???

CH = ???

DL = ???

AX = ADADh if older version of ADAPT.COM installed

Program: ADAPT is Compaq's Advanced Display Attribute Programming Tool, an

optionally-resident utility for setting display colors and cursor

size; when resident, it also includes a screen blanker

SeeAlso: AX=BFA1h,AX=BFA2h,AX=DFA5h

Index: screen saver;ADAPT

-----V-10BFA1-----

INT 10 - VIDEO - Compaq ADAPT.COM - GET ???

AX = BFA1h

Return: AX = BDBEh if supported

CH = current value of ???

DL = current value of ???

SeeAlso: AX=BFA0h,AX=BFA2h

-----V-10BFA2-----

INT 10 - VIDEO - Compaq ADAPT.COM - SET ???

AX = BFA2h

CH = new value for ???

DL = new value for ???

Return: AX = BDBEh if supported

SeeAlso: AX=BFA0h,AX=BFA1h

-----A-10C0-----

INT 10 - CU Writer v1.4 - GO TO TEXT ROW AND COLUMN

AH = C0h

???

Return: ???

SeeAlso: AH=B1h, AH=C1h, AH=C2h, AH=C3h, AH=C4h

-----V-10C000-----

INT 10 - S3 SpeedUp v3.00+ - INSTALLATION CHECK

AX = C000h

Return: AX = FFFFh if installed

DX = version number (binary, DH=major, DL=minor)

BX = resident code segment

Program: S3SPDUP is a freeware TSR by Dietmar Meschede to improve the speed of chained video modes (both VESA and VGA Mode 13) on S3-chipset video cards

SeeAlso: AX=C001h,AX=C002h,AX=C003h,AX=D000h"S3VBE"

-----V-10C001-----

INT 10 - S3 SpeedUp v3.00+ - GET ACTIVE OPTIONS

AX = C001h

Return: AX = FFFFh if function supported

BX = currently active options (see #00216)

SeeAlso: AX=C000h,AX=C002h,AX=C003h

Bitfields for S3SpeedUp options:

Bit(s) Description (Table 00216)

0 speed up banked VESA modes

1 speed up 320x200 VGA mode 13

SeeAlso: #00222

-----V-10C002-----

INT 10 - S3 SpeedUp v3.00+ - SET ACTIVE OPTIONS

AX = C002h

BX = new value for active options (see #00216)

Return: AX = FFFFh if function supported

SeeAlso: AX=C000h,AX=C001h,AX=C003h,AX=D003h"S3VBE"

-----V-10C003-----

INT 10 - S3 SpeedUp v3.10+ - GET FLAGS

AX = C003h

Return: AX = FFFFh if function supported

BX = flags

bit 0: SpeedUp is active

bit 1: Windows run after SpeedUp

SeeAlso: AX=C000h,AX=C001h,AX=C002h,AX=D005h"S3VBE"

-----V-10C004-----

INT 10 - S3 SpeedUp v3.10+ - RESERVED FUNCTIONS

AX = C004h-C0FFh

Return: AX = 5555h

SeeAlso: AX=C000h

-----A-10C1-----

INT 10 - CU Writer v1.4 - OUTPUT TEXT

AH = C1h

???

Return: ???

SeeAlso: AH=C0h,AH=C2h,AH=C3h,AH=C4h

-----A-10C2-----

INT 10 - CU Writer v1.4 - REVERSE TEXT

AH = C2h

???

Return: ???

SeeAlso: AH=C0h,AH=C1h,AH=C3h,AH=C4h

-----A-10C3-----

INT 10 - CU Writer v1.4 - TEXT BAR

AH = C3h

???

Return: ???

SeeAlso: AH=C0h,AH=C1h,AH=C4h

-----A-10C4-----

INT 10 - CU Writer v1.4 - TEXT MENU

AH = C4h

???

Return: ???

SeeAlso: AH=C0h,AH=C1h,AH=C3h,AH=D0h

-----V-10CB00-----

INT 10 - UNCHAIN - SAVE CURRENT VGA REGISTERS

AX = CB00h

Note: combined with AX=CB06h, this function permits the use of video mode

13h together with the VGA's unchained mode

SeeAlso: AX=CB06h,AX=CBFFh

-----V-10CB01-----

INT 10 - UNCHAIN - UPDATE PAGE 1 OFFSET

AX = CB01h

BX = offset of page 1

Return: nothing

Desc: inform UNCHAIN of the location of video page 1 in video memory; the

default is 0000h

SeeAlso: AX=CB02h,AX=CB03h,AX=CB04h,AX=CBFFh

-----V-10CB02-----

INT 10 - UNCHAIN - UPDATE PAGE 2 OFFSET

AX = CB02h

BX = offset of page 2

Return: nothing

Desc: inform UNCHAIN of the location of video page 2 in video memory; the

default is 3E80h (16000, for 320x200)

SeeAlso: AX=CB00h,AX=CB01h,AX=CB03h,AX=CB04h

-----V-10CB03-----

INT 10 - UNCHAIN - UPDATE PAGE 3 OFFSET

AX = CB03h

BX = offset of page 3

Return: nothing

Desc: inform UNCHAIN of the location of video page 3 in video memory; the
default is 7D00h (32000, for 320x200)

SeeAlso: AX=CB01h,AX=CB02h,AX=CB04h

-----V-10CB04-----

INT 10 - UNCHAIN - UPDATE PAGE 4 OFFSET

AX = CB04h

BX = offset of page 4

Return: nothing

Desc: inform UNCHAIN of the location of video page 4 in video memory; the
default is BB80h (48000, for 320x200)

SeeAlso: AX=CB01h,AX=CB02h,AX=CB03h,AX=CB05h

-----V-10CB05-----

INT 10 - UNCHAIN - SAVE PALETTE

AX = CB05h

Return: nothing

SeeAlso: AX=CB00h,AX=CB06h

-----V-10CB06-----

INT 10 - UNCHAIN - RESTORE VGA REGISTERS

AX = CB06h

Return: nothing

SeeAlso: AX=CB00h,AX=CB05h,AX=CBFFh

-----V-10CBFF-----

INT 10 - UNCHAIN - INSTALLATION CHECK

AX = CBFFh

Return: AX = CCBBh if installed

Program: UNCHAIN is a TSR by Colin Buckley which permits use of Borland
development tools with Mode X video in the absence of a second
monitor

SeeAlso: AX=CB00h,AX=CB01h,AX=CB06h

-----V-10CC00-----

INT 10 - VIDEO - UltraVision - GET STATUS (INSTALLATION CHECK)

AX = CC00h

SI = magic value 0000h (if checking version)

Return: CX = product signature

ABCDh UltraVision

5546h ('UF') UltraFont
AL = Ultravision extensions
00h enabled
FFh disabled
AH = card designator
BX:00F0h -> palette values (for compatibility with NEWFONT)
DX = support for high resolution modes
00h not active
01h active
SI = UltraVision version number (v1.2+), high byte=major, low byte=minor
unchanged for versions <1.2
Note: UltraFont is a simplified version of UltraVision for Toshiba;
only the signature in CX and the status in AL will be returned
SeeAlso: AX=CC01h, AX=CC02h
-----V-10CC01-----
INT 10 - VIDEO - UltraVision - DISABLE EXTENSIONS
AX = CC01h
Return: nothing
Notes: subsequent BIOS calls will be passed through to previous handler
should be followed immediately by mode set to restore normal EGA/VGA
state
SeeAlso: AX=CC02h
-----V-10CC02-----
INT 10 - VIDEO - UltraVision - ENABLE EXTENSIONS
AX = CC02h
Return: nothing
Note: should be followed immediately by mode set to restore previous
UltraVision state
SeeAlso: AX=CC01h
-----U-10CCAB-----
INT 10 - HiFont - INSTALLATION CHECK
AX = CCABh
Return: AX = ABCCh
Program: HiFont is a 8x19 font driver for standard VGA by Solar Designer
-----V-10CD00-----
INT 10 - VIDEO - UltraVision - LOAD ULTRAVISION PALETTE (color EGA,VGA)
AX = CD00h
CL = palette table number (01h-07h for v1.x, 01h-0Fh for v2+)
DS:DX -> 16-byte palette register list (colors for registers 00h-0Fh)
Return: nothing
Notes: if palette locking is in effect for the current mode, the new colors

will be displayed immediately; otherwise, the system reverts to the
default palette

palette table 0 is reserved for the default palette and cannot be set

UltraVision always sets the border color to black

SeeAlso: AX=CD01h,AX=CD02h

-----V-10CD01-----

INT 10 - VIDEO - UltraVision - SET PALETTE LOCKING STATUS (color EGA,VGA)

AX = CD01h

CL = palette locking value

00h none

01h text modes only (02h,03h)

FFh all modes (all standard color text and graphics modes)

Return: nothing

Notes: intended for video modes with 16 or fewer colors

SeeAlso: AX=1000h,AX=1002h,AX=CD00h,AX=CD03h

-----V-10CD02-----

INT 10 - VIDEO - UltraVision - GET ULTRAVISION PALETTE (EGA,VGA)

AX = CD02h

Return: CL = palette table number

DS:DX -> 17-byte palette register list (see #00217)

DS:SI -> current font names table (see #00218,#00219)

Note: only the font names are valid on monochrome EGA systems

SeeAlso: AX=1009h,AX=CD00h

Format of UltraVision palette register list:

Offset Size Description (Table 00217)

00h 16 BYTES colors for palette registers 00h through 0Fh

10h BYTE border color

Format of UltraVision v2+ current font names table:

Offset Size standard EGA HiRes EGA VGA (Table 00218)

00h 8 BYTES N/A F19 font F20 font

08h 8 BYTES F14 font F14 font F14 font

10h 8 BYTES N/A F11 font F10 font

18h 8 BYTES F8 font F8 font F8 font

Format of UltraVision v1.x current font names table:

Offset Size HiRes EGA (Table 00219)

00h 8 BYTES F19/F14 font

08h 8 BYTES F11/F8 font

-----V-10CD03-----

INT 10 - VIDEO - UltraVision - GET PALETTE LOCKING STATUS (color EGA,VGA)

AX = CD03h

Return: CL = palette locking value

00h none

01h text modes only

FFh all modes

SeeAlso: AX=CD01h

-----V-10CD04-----

INT 10 - VIDEO - UltraVision - GET UltraVision TEXT MODE (EGA,VGA)

AX = CD04h

Return: AL = mode number (see #00220)

SeeAlso: AH=0Fh,AX=CC00h,AH=CDh

(Table 00220)

Values for UltraVision video mode number:

11h 80x25

12h 80x43, 80x50

13h 80x34, 80x36

14h 80x60, 80x63

19h 94x25

1Ah 94x43, 94x50

1Bh 94x36

1Ch 94x63

21h 108x25

22h 108x43, 108x50

23h 107x34, 108x36

24h 108x60, 108x63

31h 120x25

32h 120x43, 120x50

33h 132x25

34h 132x44, 132x50

39h 120x36

3Ah 120x63

3Bh 132x36

3Ch 132x60

Index: video modes;UltraVision

-----V-10CD05-----

INT 10 - VIDEO - UltraVision - SET CURSOR TYPE (EGA,VGA)

AX = CD05h

CL = type

00h line cursor

FFh box cursor

Return: nothing

Note: sets default cursor type for text-based programs

SeeAlso: AH=01h,AX=CD06h

-----V-10CD06-----

INT 10 - VIDEO - UltraVision - GET CURSOR TYPE (EGA,VGA)

AX = CD06h

Return: CL = type

00h line cursor

FFh box cursor

SeeAlso: AH=03h,AX=CD05h

-----V-10CD07-----

INT 10 - VIDEO - UltraVision v1.2+ - SET UNDERLINE STATUS (EGA,VGA)

AX = CD07h

CL = hardware underline status (see #00221)

BL = foreground color for normal text (FFh = current)

BH = foreground color for bright text (FFh = current)

Return: CL = hardware underline status

BL = current foreground color for normal text

BH = current foreground color for bright text

Notes: when underline or strikeout is enabled in color text modes, the specified colors will be assigned temporarily to colors 01h and 09h, allowing affected text to match non-underlined text. The color remapping uses values from the current onscreen palette regardless of the palette locking status (see AX=CD01h)

specify the standard colors (BL=01h,BH=09h) to enable underline or strikeout without color remapping

SeeAlso: AX=CD08h

(Table 00221)

Values for hardware underline status:

00h off (color systems only)

01h underline below characters

02h strike through characters

-----V-10CD08-----

INT 10 - VIDEO - UltraVision v1.2+ - GET UNDERLINE STATUS (EGA,VGA)

AX = CD08h

Return: CL = hardware underline status (see #00221)

BL = foreground color for normal text

BH = foreground color for bright text

Note: only CL is valid on monochrome EGA systems

SeeAlso: AX=CD07h

-----V-10CD0F-----

INT 10 - VIDEO - UltraVision - GET POINTER TO ??? (EGA,VGA)

AX = CD0Fh

Return: DS:DI -> pointer to ???

Note: This gets called by DR DOS "Panther" SECURITY. SECURITY also issues a number of calls to the other UltraVision functions in the AH=CCh and AH=CDh range, and carefully checks signatures.

SeeAlso: AX=1100h,AX=1103h

-----V-10CD10-----

INT 10 - VIDEO - UltraVision - LOAD USER FONT (EGA,VGA)

AX = CD10h

BH = bytes per character (08h,0Ah,0Bh,0Eh,13h,14h)

CX = ABCDh load 9xN alternate font (v2+)

else number of characters to load

DX = character offset into font table

DS:SI -> 8-byte ASCII font name

ES:BP -> font definitions

Return: AX = FFFFh if invalid font parameters

Notes: loads the designated characters into UltraVision's resident font area should be followed by a video mode set to reload character generator

SeeAlso: AX=1100h,AX=1103h

-----V-10CD-----

INT 10 - VIDEO - UltraVision - SET ULTRAVISION TEXT MODE (EGA,VGA)

AH = CDh

AL = text mode number (see #00220)

Return: AX = CDCDh if invalid mode

SeeAlso: AX=CD04h

-----A-10D0-----

INT 10 - CU Writer v1.4 - LOAD PICTURE

AH = D0h

???

Return: ???

SeeAlso: AH=B0h"CU Writer",AH=C4h

-----V-10D0-----

INT 10 U - VIDEO - HP 100LX/200LX - SET ZOOM MODE

AH = D0h

AL = zoom mode

02h 80x25 mono

03h 80x25 color

80h 64x18 mono

81h 64x18 color
82h 40x25 mono
83h 40x25 color
84h 40x16 mono
85h 40x16 color

Return: nothing

Note: zoom mode can only be changed within zoom modes of the same color scheme; if needed, set to mono/color with AH=00h,AL=02h/03h; with mono video modes AL=07h or AL=21h only 80x25 and 40x16 will work the current zoom mode is stored in the BIOS data area at 0040h:009Fh

SeeAlso: AH=D1h,AH=D4h

-----V-10D000-----

INT 10 - S3VBE/Core2.0 v3.00+ - INSTALLATION CHECK

AX = D000h

Return: AX = FFFFh if installed

BX = resident code segment

DX = version (binary, DH = major, DL = minor)

Program: S3VBE/Core2.0 is a freeware TSR by Dietmar Meschede to provide VESA 2.0 services on S3-based video cards with VESA 1.2 BIOS

SeeAlso: AX=D001h"S3VBE",AX=D002h"S3VBE",AX=D003h"S3VBE",AX=D005h"S3VBE"

-----V-10D001-----

INT 10 - S3VBE/Core2.0 v3.00+ - GET ACTIVE OPTIONS

AX = D001h

Return: AX = FFFFh if supported

BX = currently active options (see #00222)

SeeAlso: AX=D000h"S3VBE",AX=D002h"S3VBE"

Bitfields for S3VBE options:

Bit(s) Description (Table 00222)

0 VESA VBE v2.0 extensions enabled

1 linear frame buffer enabled

2 low-resolution video mode support enabled

8 never clear video memory during VBE mode set (v3.10+)

---debug options---

12 report VBE version 1.2 (v3.10+)

13 always fail AX=4F0Ah (v3.10+)

14 don't copy video mode list (v3.12+)

SeeAlso: #00216

-----V-10D002-----

INT 10 - S3VBE/Core2.0 v3.00+ - SET ACTIVE OPTIONS

AX = D002h


```
    BX = new active options (see #00222)
Return: AX = FFFFh if supported
SeeAlso: AX=D000h"S3VBE",AX=D001h"S3VBE"
-----V-10D003-----
INT 10 U - S3VBE/Core2.0 v3.10+ - GET VBE/Core CAPABILITIES
    AX = D003h
Return: AX = FFFFh if supported
    BX = capabilities
    bit 0: SpeedUp = activate/deactivate linear addressing at
    A0000h for VBE functions 04h/05h
SeeAlso: AX=D000h"S3VBE",AX=D001h"S3VBE",AX=D004h"S3VBE",AX=D005h"S3VBE"
-----V-10D004-----
INT 10 U - S3VBE/Core2.0 v3.10+ - ACTIVATE SPEED-UP
    AX = D004h
Return: AX = FFFFh if supported
    BX = status
    0000h SpeedUp activated
    0001h SpeedUp not possible (wrong memory organization for mode)
    0002h SpeedUp not possible (linear frame buffer active)
Note: called by S3 SpeedUp (see AX=C000h)
SeeAlso: AX=C000h"SpeedUp",AX=D000h"S3VBE",AX=D003h"S3VBE",AX=D005h"S3VBE"
-----V-10D005-----
INT 10 U - S3VBE/Core2.0 v3.10+ - DEACTIVATE SPEED-UP
    AX = D005h
Return: AX = FFFFh if supported
    BX = status
    0000h SpeedUp deactivated
    0001h SpeedUp not possible (wrong memory organization for mode)
    0002h SpeedUp not possible (linear frame buffer active)
Notes: called by S3 SpeedUp (see AX=C000h)
    functions 06h-FFh (e.g. AX=D006h-D0FFh) are considered reserved by
    v3.10+ and return AX=5555h
SeeAlso: AX=C000h"SpeedUp",AX=D000h"S3VBE",AX=D003h"S3VBE",AX=D004h"S3VBE"
-----V-10D1-----
INT 10 U - VIDEO - HP 100LX/200LX - INTERNAL - ???
    AH = D1h
    AL = 01h
Return: ???
Note: called by AH=D0h
SeeAlso: AH=D0h,AH=D4h
-----V-10D4-----
```

INT 10 U - VIDEO - HP 100LX/200LX - INTERNAL - ???

AH = D4h

AL = 29h

Return: ???

Note: called by AH=D0h

SeeAlso: AH=D0h,AH=D1h

-----V-10D5-----

INT 10 - Netroom SCRNCCLK - ???

AH = D5h

???

Return: ???

Program: SCRNCCLK is a "cloaked" screen accelerator included with Netroom

-----t-10DAAD-----

INT 10 - TSRUNIT v1.10 - INSTALLATION CHECK

AX = DAADh

BX = check signature (different for each TSR)

CX = 0000h

Return: CX = return signature (nonzero) if installed

ES = program segment prefix

Program: TSRUNIT is a Turbo Pascal unit for creating TSRs by Nir Sofer

-----V-10DFA5-----

INT 10 U - VIDEO - Compaq ADAPT.COM - GET ??? DATA AREA

AX = DFA5h

Return: AX = BDBFh if supported

ES:DI -> ??? data area

BX = ES

SeeAlso: AX=BFA0h

-----V-10EE-----

INT 10 - VIDEO - IBM "Private" Function

AH = EEh

SeeAlso: INT 13/AH=FFh"IBM"

-----V-10EF-----

INT 10 - VIDEO - MSHERC.COM - GET VIDEO ADAPTER TYPE AND MODE

AH = EFh

Return: DL = video adapter type

00h original Hercules

01h Hercules Plus (port 03BAh reads x001xxxxx)

02h Hercules InColor (port 03BAh reads x101xxxxx)

FFh not a Hercules-compatible card (port 03BAh bit 7 not pulsing)

DH = memory mode byte

00h "half" mode

01h "full" mode

FFh not a Hercules-compatible card

Program: MSHERC.COM/QBHERC.COM is a support program for the Microsoft Quick languages which makes their graphics libraries compatible with a Hercules card by adding video modes 08h and 88h, and supporting text in the new graphics modes.

Notes: while in mode 08h or 88h, INT 10 supports the Hercules card much like a CGA.

MSHERC performs an installation check by setting DL=FFh and testing whether it has been changed on return, which causes it to reinstall itself when no HGC is present (or HGC emulation has temporarily been disabled); a better installation check would be to use DX=80FFh and check whether DX has been changed reportedly returns DH=00h on some not-entirely-Hercules-compatible cards

Index: installation check;MSHERC

-----V-10F0-----

INT 10 - EGA Register Interface Library - READ ONE REGISTER

AH = F0h

BL = register number

BH = 00h

DX = group index (see #00223)

Return: BL = data

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2 compatibility box, and others; it is used for software virtualization of write-only registers on an EGA video adapter, so that multiple programs may peacefully coexist without clobbering each other's display settings

SeeAlso: AH=F1h"EGA",AH=F2h"EGA",AH=FAh"EGA",INT 2F/AX=BC00h

(Table 00223)

Values for group index:

Pointer/data chips

00h CRT Controller (25 reg) 3B4h mono modes, 3D4h color modes

08h Sequencer (5 registers) 3C4h

10h Graphics Controller (9 registers) 3CEh

18h Attribute Controller (20 registers) 3C0h

Single registers

20h Miscellaneous Output register 3C2h

28h Feature Control register (3BAh mono modes, 3DAh color modes)

30h Graphics 1 Position register 3CCh

38h Graphics 2 Position register 3CAh

-----V-10F0-----

INT 10 - VHRBIOS.SYS - INSTALLATION CHECK

AH = F0h

Return: BX = 4F4Bh ('OK')

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"

black&white A4/portrait monitor

SeeAlso: AH=F1h"VHRBIOS.SYS",AH=F2h"VHRBIOS.SYS",AH=F4h"VHRBIOS.SYS"

SeeAlso: AH=96h"VHRBIOS.SYS"

-----V-10F1-----

INT 10 - EGA Register Interface Library - WRITE ONE REGISTER

AH = F1h

DX = group index (see #00223)

if single register:

BL = value to write

otherwise

BL = register number

BH = value to write

Return: BL = data

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2

compatibility box, and others

SeeAlso: AX=7F05h,AH=F0h"EGA",AH=F3h"EGA",AH=FAh"EGA"

-----V-10F1-----

INT 10 - VHRBIOS.SYS - SET REVERSE VIDEO

AH = F1h

AL = new video state

(bit 5 set for black text on white, clear for white on black)

Return: ???

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"

black&white A4/portrait monitor

SeeAlso: AH=F0h"VHRBIOS.SYS",AH=F3h"VHRBIOS.SYS"

-----V-10F2-----

INT 10 - EGA Register Interface Library - READ REGISTER RANGE

AH = F2h

CH = starting register number

CL = number of registers (>1)

DX = group index (00h,08h,10h,18h) (see #00223)

ES:BX -> buffer, CL bytes

Return: nothing

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2

compatibility box, and others

SeeAlso: AH=F0h"EGA",AH=F3h"EGA",AH=FAh"EGA"

-----V-10F2-----

INT 10 - VHRBIOS.SYS - ???

AH = F2h

Return: ???

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"

black&white A4/portrait monitor

SeeAlso: AH=F0h"VHRBIOS.SYS",AH=F4h"VHRBIOS.SYS"

-----V-10F3-----

INT 10 - EGA Register Interface Library - WRITE REGISTER RANGE

AH = F3h

CH = starting register

CL = number of registers (>1)

DX = group index (00h,08h,10h,18h) (see #00223)

ES:BX -> buffer, CL bytes

Return: nothing

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2 compatibility box, and others

SeeAlso: AX=7F05h, AH=F1h"EGA", AH=F2h"EGA", AH=F4h"EGA"

-----V-10F3-----

INT 10 - VHRBIOS.SYS - SWITCH BETWEEN DUAL MONITORS???

AH = F3h

Return: ???

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"

black&white A4/portrait monitor

SeeAlso: AH=F0h"VHRBIOS.SYS",AH=F4h"VHRBIOS.SYS",AH=F6h"VHRBIOS.SYS"

-----V-10F4-----

INT 10 - EGA Register Interface Library - READ REGISTER SET

AH = F4h

CX = number of registers to read (>1)

ES:BX -> table of register records (see #00224)

Return: register values in table filled in

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2 compatibility box, and others

SeeAlso: AH=F0h"EGA",AH=F2h"EGA",AH=F5h"EGA"

Format of EGA RIL entries in table of register records:

Offset Size Description (Table 00224)

00h WORD group index

Pointer/data chips

00h CRTIC (3B4h mono modes, 3D4h color modes)

08h Sequencer 3C4h
10h Graphics Controller 3CEh
18h Attribute Controller 3C0h

Single registers

20h Miscellaneous Output register 3C2h
28h Feature Control register (3BAh mono modes, 3DAh color)
30h Graphics 1 Position register 3CCh
38h Graphics 2 Position register 3CAh

02h BYTE register number (0 for single registers)

03h BYTE register value

-----V-10F4-----

INT 10 - VHRBIOS.SYS - GET VERSION

AH = F4h

Return: AX = driver version (AH = major, AL = minor)

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"

black&white A4/portrait monitor

SeeAlso: AH=F0h"VHRBIOS.SYS",AH=F5h"VHRBIOS.SYS"

-----V-10F5-----

INT 10 - EGA Register Interface Library - WRITE REGISTER SET

AH = F5h

CX = number of registers to write (>1)

ES:BX -> table of records (see #00224)

Return: nothing

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2

compatibility box, and others

SeeAlso: AX=7F05h,AH=F1h"EGA",AH=F3h"EGA",AH=F4h"EGA"

-----V-10F5-----

INT 10 - VHRBIOS.SYS - GET VENDOR ID

AH = F5h

Return: AXBX = vendor ID (4D44h:5349h = 'MDSI' for Micro Display Systems Inc.)

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"

black&white A4/portrait monitor

SeeAlso: AH=F0h"VHRBIOS.SYS",AH=F4h"VHRBIOS.SYS",AH=F6h"VHRBIOS.SYS"

-----V-10F6-----

INT 10 - EGA Register Interface Library - REVERT TO DEFAULT REGISTERS

AH = F6h

Return: nothing

Note: provided by the Microsoft Mouse driver, OS/2 compatibility box, and

others

SeeAlso: AH=F5h"EGA",AH=F7h"EGA"

-----V-10F6-----

INT 10 - VHRBIOS.SYS - GET INFO

AH = F6h

AL = what to get

00h device driver state

Return: AX = device driver state

01h video mode info

Return: AL = video mode

DH = screen height in rows

DL = screen width in columns

Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"
black&white A4/portrait monitor

SeeAlso: AH=F0h"VHRBIOS.SYS",AH=F4h"VHRBIOS.SYS",AH=96h"VHRBIOS.SYS

-----V-10F7-----

INT 10 - EGA Register Interface Library - DEFINE DEFAULT REGISTER TABLE

AH = F7h

DX = port number

Pointer/data chips

00h CRTC (3B4h mono modes, 3D4h color modes)

08h Sequencer 3C4h

10h Graphics Controller 3CEh

18h Attribute Controller 3C0h

Single registers

20h Miscellaneous Output register 3C2h

28h Feature Control register (3BAh mono modes, 3DAh color modes)

30h Graphics 1 Position register 3CCh

38h Graphics 2 Position register 3CAh

ES:BX -> table of one-byte entries, one byte to be written to each
register

Return: nothing

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2
compatibility box, and others

SeeAlso: AH=F0h"EGA",AH=F6h"EGA"

-----V-10FA--BX0000-----

INT 10 - EGA Register Interface Library - INTERROGATE DRIVER

AH = FAh

BX = 0000h

Return: BX = 0000h if RIL driver not present

ES:BX -> EGA Register Interface version number, if present:

byte 0 = major release number

byte 1 = minor release number

Note: the RIL is provided by EGA.SYS, the Microsoft Mouse driver, the OS/2

compatibility box, and others

SeeAlso: AH=F0h"EGA",AH=F6h"EGA",INT 2F/AX=BC00h

-----K-10FA-----

INT 10 - FASTBUFF.COM - INSTALLATION CHECK

AH = FAh

Return: AX = 00FAh if installed

ES = segment of resident code

Program: FASTBUFF.COM is a keyboard speedup/screen blanking utility by David Steiner

Index: screen saver;FASTBUFF

-----V-10FE-----

INT 10 - TopView - GET SHADOW BUFFER

AH = FEh

ES:DI -> assumed video buffer

B800h:0000h color text/CGA graphics, B000h:0000h mono text,
or A000h:0000h EGA/VGA graphics (RSIS environments only)

Return: ES:DI -> actual video buffer for calling process

Desc: Determine the address of the virtual screen to which the program should write instead of the actual video memory; this permits programs to be multitasked without interfering with each other's output, and allows memory managers to move the video memory to permit larger programs to be loaded.

Notes: if no multitasker or RSIS-compliant environment is installed, ES:DI is returned unchanged; RSIS is the Relocated Screen Interface Specification

for display pages other than 0, use AH=05h and AH=0Fh to determine whether a particular page exists

TopView requires a call to AH=FFh to notify it that the screen has changed; DESQview will check for changes itself until the first call to AH=FFh

SeeAlso: AH=05h,AX=5201h,AH=FFh,INT 15/AX=1024h,INT 21/AH=2Bh"DESQview"

SeeAlso: INT 21/AH=ECh"DoubleDOS"

-----V-10FF-----

INT 10 - TopView - UPDATE SCREEN FROM SHADOW BUFFER

AH = FFh

CX = number of consecutive changed characters

ES:DI -> first changed character in shadow buffer

Return: nothing

Notes: avoid CX=0000h

DESQview will discontinue the automatic screen updating initiated by AH=FEh after this call

not supported (ignored) by DESQview/X 1.0x

SeeAlso: AH=93h,AH=FEh

-----E-10FF-----

INT 10 - DJ GO32.EXE 80386+ DOS extender - VIDEO EXTENSIONS

AH = FFh

AL = video mode (see #00225)

Program: GO32.EXE is a DOS extender included as part of the 80386 port of the
GNU C/C++ compiler by DJ Delorie and distributed as DJGPP

SeeAlso: AH=00h,INT 21/AH=FFh"GO32"

(Table 00225)

Values for GO32 video mode number:

00h 80x25 text

01h default text

02h CXxDX text

03h biggest text

04h 320x200 graphics

05h default graphics

06h CXxDX graphics

07h biggest non-interlaced graphics

08h biggest graphics

Index: video modes;GO32

-----V-10FF-----

INT 10 - VIDEO - Oak VGA BIOS v1.02+ - SET EMULATION

AH = FFh

AL = emulation

43h ('C') CGA emulation

45h ('E') EGA emulation

4Dh ('M') Hercules emulation

56h ('V') VGA emulation

ES:DI -> signature string "Calamity"

Return: VGA switched to suggested mode

SeeAlso: AH=00h,AX=007Fh/BH=00h,AX=007Fh/BH=02h,AX=5F01h

-----R-10FF00-----

INT 10 - CARBON COPY PLUS v5.0 - CHECK IF CC CONNECTED TO CCHELP

AX = FF00h

Return: BL = state

00h not connected

01h connected

SeeAlso: AX=FF01h,AX=FF02h

-----R-10FF01-----

INT 10 - CARBON COPY PLUS v5.0 - DISCONNECT AND RESET LINE

AX = FF01h

Return: nothing

SeeAlso: AX=FF00h,AX=FF02h

-----R-10FF02-----

INT 10 - CARBON COPY PLUS v5.0 - GET LAST PHONE NUMBER DIALED

AX = FF02h

Return: ES:DI -> ASCIZ phone number

SeeAlso: AX=FF00h,AX=FF01h

-----C-11-----

INT 11 - CPU-generated (80486+) - ALIGNMENT CHECK

Desc: automatically generated by the CPU when the AC flag is set, the current privilege level is 3, and a misaligned memory access (WORD not on an even address or DWORD not on a multiple of 4) is made

Note: not all V86 monitors allow the AC flag to be set, such as Turbo

Debugger 386

SeeAlso: INT 12"CPU"

-----B-11-----

INT 11 - BIOS - GET EQUIPMENT LIST

Return: (E)AX = BIOS equipment list word (see #00226,#03215 at INT 4B"Tandy")

Note: since older BIOSes do not know of the existence of EAX, the high word of EAX should be cleared before this call if any of the high bits will be tested

SeeAlso: INT 4B"Tandy 2000",MEM 0040h:0010h

Bitfields for BIOS equipment list:

Bit(s) Description (Table 00226)

0 floppy disk(s) installed (number specified by bits 7-6)

1 80x87 coprocessor installed

3-2 number of 16K banks of RAM on motherboard (PC only)

number of 64K banks of RAM on motherboard (XT only)

2 pointing device installed (PS)

3 unused (PS)

5-4 initial video mode

00 EGA, VGA, or PGA

01 40x25 color

10 80x25 color

11 80x25 monochrome

7-6 number of floppies installed less 1 (if bit 0 set)

8 DMA support installed (PCjr, Tandy 1400LT)

DMA support *not* installed (Tandy 1000's)

```
11-9 number of serial ports installed
12 game port installed
13 serial printer attached (PCjr)
   internal modem installed (PC/Convertible)
15-14 number of parallel ports installed
---Compaq, Dell, and many other 386/486 machines--
23 page tables set so that Weitek coprocessor addressable in real mode
24 Weitek math coprocessor present
---Compaq Systempro---
25 internal DMA parallel port available
26 IRQ for internal DMA parallel port (if bit 25 set)
   0 = IRQ5
   1 = IRQ7
28-27 parallel port DMA channel
   00 DMA channel 0
   01 DMA channel 0 ???
   10 reserved
   11 DMA channel 3
```

Notes: Some implementations of Remote (Initial) Program Loader (RPL/RIPL) don't set bit 0 to indicate a "virtual" floppy drive, although the RPL requires access to its memory image through a faked drive A:. This may have caused problems with releases of DOS 3.3x and earlier, which assumed A: and B: to be invalid drives then and would discard any attempts to access these drives. Implementations of RPL should set bit 0 to indicate a "virtual" floppy.

The IBM PC DOS 3.3x-2000 IBMBIO.COM contains two occurrences of code sequences like:

```
INT 11h
JMP SHORT skip
DB 52h,50h,53h; "RPS"
skip: OR AX,1
TEST AX,1
```

While at the first glance this seems to be a bug since it just wastes memory and the condition is always true, this could well be a signature for an applicable patch to stop it from forcing AX bit 0 to be always on. MS-DOS IO.SYS does not contain these signatures, however.

BUGs: Some old BIOSes didn't properly report the count of floppy drives installed to the system. In newer systems INT 13h/AH=15h can be used to retrieve the number of floppy drives installed.

Award BIOS v4.50G and v4.51PG erroneously set bit 0 even if there are

no floppy drives installed; use two calls to INT 13/AH=15h to
determine whether any floppies are actually installed
SeeAlso: INT 12"BIOS",#03215 at INT 4B"Tandy 2000"
-----d-11----SI6A6A-----
INT 11 - Columbia Data Products Standard Device Level Protocol (SDLP) 1.6
SI = 6A6Ah
AH = command (see #00227)
AL = SCSI Addressing (see #00228)
Return: CF clear if successful
DI = 6A6Ah if AH=01h on entry (maybe for all functions???)
AH = ??? for command 01h
CF set on error
AL = error code
SeeAlso: INT 21/AX=4402h"ASPI"

(Table 00227)

Values for SDLP command:

00h SDLP initialization
01h SDLP System Identify
02h simple read sectors
03h simple write sectors
04h simple verify sectors/seek to sector
05h get device size/type
06h ready unit
07h format unit
08h diagnostics
09h rewind
0Ah erase
0Bh write filemarks
0Ch space
0Dh prevent/allow media removal
0Eh load/unload media
0Fh reserved - returns good status
10h set block size
11h write setmark
12h set error level
13h get address of Request Sense Buffer
14h get SDLP error via Request Sense
F0h Vendor Unique Function (WD7000-FASST2 only)
FDh reset current SCSI HAC
FEh get/set current SCSI HAC

FFh execute SCSI command

Bitfields for SDLP SCSI addressing:

Bit(s) Description (Table 00228)

7-6 Host Adapter

5-3 SCSI Target ID

2-0 SCSI Target LUN (logical unit number)

-----V-110225BX6900-----

INT 11 - Blank-It Screen Blanker - INSTALLATION CHECK

AX = 0225h

BX = 6900h

Return: BL = 23h

ES:DI -> ASCIZ "BLNKIT"

Program: Blank-It is a resident screen blanker by Rhode Island Soft Systems,
Inc.

SeeAlso: AX=0225h/BX=6902h,AX=0225h/BX=6908h,AX=0225h/BX=6909h,INT 14/AX=AA01h

Index: screen saver;Blank-It

-----V-110225BX6901-----

INT 11 - Blank-It Screen Blanker - SET TIMEOUT FOR SCREEN BLANKING

AX = 0225h

BX = 6901h

CX = timeout in timer ticks (18.2/second) or 0000h to disable timeout
largest value is 59 minutes (FBACH or 64428)

Return: CF clear if successful

DI = 6A6Ah (possibly also 6A6Ah for all following functions)

AH = ???

CF set on error

AL = error code

SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6904h

Index: screen saver;Blank-It

-----V-110225BX6902-----

INT 11 - Blank-It Screen Blanker - ENABLE THE SOFTWARE

AX = 0225h

BX = 6902h

Return: CF clear if successful

CF set on error

AL = error code

SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6903h

Index: screen saver;Blank-It

-----V-110225BX6903-----

INT 11 - Blank-It Screen Blanker - DISABLE THE SOFTWARE

```
AX = 0225h
BX = 6903h
Return: CF clear if successful
CF set on error
AL = error code
SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6902h
Index: screen saver;Blank-It
-----V-110225BX6904-----
INT 11 - Blank-It Screen Blanker - GET BLANKING TIMEOUT
AX = 0225h
BX = 6904h
Return: CF clear if successful
BX = timeout (see AX=0225h/BX=6901h)
CF set on error
AL = error code
SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6901h
Index: screen saver;Blank-It
-----V-110225BX6905-----
INT 11 - Blank-It Screen Blanker - ENABLE WINDOWS COMPATIBILITY MODE
AX = 0225h
BX = 6905h
Return: CF clear if successful
CF set on error
AL = error code
SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6906h
Index: screen saver;Blank-It
-----V-110225BX6906-----
INT 11 - Blank-It Screen Blanker - DISABLE WINDOWS COMPATIBILITY MODE
AX = 0225h
BX = 6906h
Return: CF clear if successful
CF set on error
AL = error code
Program: Blank-It is a resident screen blanker by Rhode Island Soft Systems,
Inc.
SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6905h
Index: screen saver;Blank-It
-----V-110225BX6907-----
INT 11 - Blank-It Screen Blanker - UNBLANK THE SCREEN
AX = 0225h
BX = 6907h
```

Return: CF clear if successful

CF set on error

AL = error code

SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6908h

Index: screen saver;Blank-It

-----V-110225BX6908-----

INT 11 - Blank-It Screen Blanker - BLANK THE SCREEN

AX = 0225h

BX = 6908h

Return: CF clear if successful

CF set on error

AL = error code

SeeAlso: AX=0225h/BX=6900h,AX=0225h/BX=6907h

Index: screen saver;Blank-It

-----V-110225BX6909-----

INT 11 - Blank-It Screen Blanker - SET HOTKEY FOR MANUAL BLANKING

AX = 0225h

BX = 6909h

CL = key scan code (see #00229)

Return: CF clear if successful

CF set on error

AL = error code

Program: Blank-It is a resident screen blanker by Rhode Island Soft Systems,
Inc.

SeeAlso: AX=0225h/BX=6900h

Index: screen saver;Blank-It|hotkeys;Blank-It

(Table 00229)

Values for Blank-It hotkey scan code:

00h No hot key

1Dh Left CTRL

2Ah Left Shift

36h Right Shift

57h F11

58h F12

SeeAlso: #00006

-----G-1105D7-----

INT 11 CU - Borland C++ IDE - INSTALLED CALLOUT

AX = 05D7h

BX = product ID (0088h)

Note: called by the BC++ IDE when an application calls

INT 12/AX=05D7h/BX=05D7h

SeeAlso: INT 12/AX=05D7h/BX=05D7h

Index: installation check;Borland C++ IDE

-----F-1177-----

INT 11 - RainbowFAX v1.3 - SFENGINE API - OPERATIONAL CONTROL

AH = 77h

AL = subfunction

01h request SFENGINE start

Return: AX = 0001h

02h check if SFENGINE started

Return: AX = 0000h or 0001h

03h request SFENGINE stop

Return: AX = 0001h

04h check if SFENGINE stopped

Return: AX = 0000h or 0001h

05h installation check

Return: AX = 0001h

06h uninstall???

BX:DX -> return address for successful uninstall???

Return: (at caller's address)

AX = 0000h

else

Return: AX = FFFFh (invalid subfunction)

Return: ES:DX -> ASCIZ signature string "SFAX ENGINE V1.0"

followed by internal data area???

SeeAlso: AH=78h,AH=79h,AH=7Ah,AH=7Ch

Index: installation check;RainbowFAX|installation check;SFENGINE

Index: uninstall;RainbowFAX|uninstall;SFENGINE

-----F-1178-----

INT 11 - RainbowFAX v1.3 - SFENGINE API - ???

AH = 78h

AL = subfunction

01h set ???

BX = new state for ???

0000h ???

else ???

Return: AX = 0001h

02h set ???

BX = ???

CX = ???

Return: AX = 0001h


```
    03h set ???
BX = new state for ???
    0000h ???
    else ???
Return: AX = 0001h
    else
Return: AX = FFFFh (invalid subfunction)
SeeAlso: AH=77h,AH=79h,AH=7Ah,AH=7Ch
-----F-1179-----
INT 11 - RainbowFAX v1.3 - SFENGINE API - ???
AH = 79h
AL = subfunction
    01h set ??? flag
Return: AX = previous state (0000h already set, 0001h clear)
    02h clear ??? flag (refer to subfunc 01h)
Return: AX = 0001h
    03h set ??? flag (different from subfn 02h or 04h)
Return: AX = 0001h
    04h clear ??? flag (different from subfn 02h or 03h)
Return: AX = 0001h
    else
Return: AX = FFFFh (invalid subfunction)
SeeAlso: AH=77h,AH=78h,AH=7Ah,AH=7Ch
-----F-117A-----
INT 11 - RainbowFAX v1.3 - SFENGINE API - ???
AH = 7Ah
AL = subfunction
    01h set ??? flag
Return: AX = previous state (0000h already set, 0001h clear)
    02h clear ??? flag
Return: AX = 0001h
    03h set ??? flag (different from subfn 02h)
Return: AX = 0001h
    else
Return: AX = FFFFh (invalid subfunction)
SeeAlso: AH=77h,AH=78h,AH=79h,AH=7Ch
-----F-117C01-----
INT 11 - RainbowFAX v1.3 - SFENGINE API - SET ??? FLAG
AX = 7C01h
Return: AX = 0001h
SeeAlso: AH=77h,AH=78h,AH=79h,AH=7Ah,AX=7C02h,AX=7C03h
```

-----F-117C02-----

INT 11 - RainbowFAX v1.3 - SFENGINE API - CLEAR ??? FLAG

AX = 7C02h

Return: AX = 0001h

SeeAlso: AH=77h,AH=78h,AH=79h,AH=7Ah,AX=7C01h,AX=7C03h

-----F-117C03-----

INT 11 - RainbowFAX v1.3 - SFENGINE API - UNSUPPORTED FUNCTIONS

AX = 7C03h-7CFFh

Return: AX = FFFFh

SeeAlso: AH=77h,AX=7C01h,AX=7C02h

-----S-11BC--DX1954-----

INT 11 - BNU FOSSIL - INSTALLATION CHECK

AH = BCh

DX = 1954h

Return: AX = 1954h

ES:DX -> entry point of driver (instead of INT 14)

SeeAlso: INT 14/AH=04h"FOSSIL"

-----d-11FF--SI6A6A-----

INT 11 - WD7000 SDLP interface - EXECUTE GENERIC SCSI COMMAND

AH = FFh

SI = 6A6Ah

AL = SCSI Addressing (see #00230)

CX = bytes of data to be transmitted (max FFF0h)

DH = 00h

DL = length of SCSI Command Descriptor Block

DS:DI -> SCSI Command Descriptor Block

ES:BX -> data buffer

Return: CF set on error

AL = error code

CF clear if successful

Note: because of busmaster operations with WD7000FASST avoid accessing
video memory directly; check 386 memory manager for VDS support.

The WD7000XTAT works with programmed IO and does not have this
limitation.

SeeAlso: INT 21/AX=4402h"ASPI",INT 2F/AX=7F01h

Bitfields for SDLP SCSI addressing:

Bit(s) Description (Table 00230)

2-0 SCSI Target LUN (logical unit number)

5-3 SCSI Target ID

7 write flag, set for write operations, clear otherwise

-----T-11FFFEEXFFFE-----

INT 11 - BACK&FORTH (before v1.62) API

AX = FFFEh

CX = FFFEh

BX = function

00h installation check

Return: AX = installation state

0001h BNFHIGH and BNFLOW both loaded

0003h only BNFHIGH loaded

else neither loaded

01h ???

Return: DX:AX -> ???

02h ???

03h ???

04h ???

05h ??? switches current PSP segment and stack if BNFLOW has not
yet announced itself installed

06h ???

Return: AX = ???

SeeAlso: INT 12/AX=FFFEh

Index: installation check;BACK&FORTH

-----B-12-----

INT 12 - BIOS - GET MEMORY SIZE

Return: AX = kilobytes of contiguous memory starting at absolute address 00000h

Note: this call returns the contents of the word at 0040h:0013h; in PC and

XT, this value is set from the switches on the motherboard

SeeAlso: INT 11"BIOS",INT 2F/AX=4A06h,INT 4C"Tandy 2000",MEM 0040h:0013h

-----C-12-----

INT 12 - CPU-generated (Pentium +) - MACHINE CHECK EXCEPTION

Notes: Intel documents this interrupt as CPU model-dependent

for current Pentium processors, the reason for the machine check
exception may be read from model-specific registers 00h and 01h
(described, for example, in Christian Ludloff's 4P package)

for Pentium Pro/II processors, the reason may be read from the
MCG_STATUS MSR (see MSR 0000017Ah)

this exception is enabled by bit 6 of CR4

SeeAlso: INT 11"CPU",MSR 00000000h,MSR 00000001h,MSR 0000017Ah

-----K-12----CX1806-----

INT 12 - KEYBUI v2.0+ - INSTALLATION CHECK

CX = 1806h

Return: AX = kilobytes of contiguous memory starting at absolute address 00000h

```
CX = 1960h if installed
Program: KEYBUI is a resident keyboard driver by Johan Zwiekhorst which allows
    accented characters and box drawing on standard QWERTY keyboards; it
    also provides break-to-DOS and screen blanking capabilities
SeeAlso: INT 14/AX=AA01h
Index:  screen saver;KEYBUI
-----d-12----CX1807-----
INT 12 - PARKER v2.0+ - INSTALLATION CHECK
    CX = 1807h
Return: AX = kilobytes of contiguous memory starting at absolute address 00000h
    CX = 1961h if installed
Program: PARKER is an optionally-resident hard disk parking program by Johan
    Zwiekhorst
-----G-1205D7BX05D7-----
INT 12 U - Borland C++ IDE - INSTALLATION CHECK
    AX = 05D7h (1495d)
    BX = 05D7h
Note: the BC++ IDE will call INT 11/AX=05D7h/BX=0088h if it is loaded
SeeAlso: INT 11/AX=05D7h
-----v-124350BX4920-----
INT 12 C - CPI-standard virus - "FRIEND" CHECK
    AX = 4350h
    BX = 4920h
    CX = AB46h
    DX = 554Eh
Return: if friendly (not to be infected)
    CX:DX -> ASCIZ identity code (changes yearly)
SeeAlso: INT 13/AX=EC00h"VIRUS", INT 13/AX=5001h, INT 21/AX=0B56h
-----T-12FFFE CXFFFE-----
INT 12 - Back&Forth v1.62+ - API
    AX = FFFEh
    CX = FFFEh
    BX = function
        00h installation check
Return: AX = 0001h installed
        else not loaded
        01h (reserved)
        02h build program ID list (shareware Back&Forth)
ES:DI -> buffer of at least 100 bytes, to be filled with words
Return: AX = number of programs defined
    ES:DI buffer filled with AX words
```

02h get memory statistics (Back&Forth Professional)
Return: AX = available swap memory, KBytes
BX = maximum task size, KBytes
DX = fixed overhead per task, excluding video/macro storage

03h switch to specified task (task need not be open yet)
DX = two-letter program ID
Return: AX = status
0000h if task undefined
0001h task switch will occur when safe

04h (reserved)
05h (reserved)
06h get version (documented only for Back&Forth Professional)
Return: AX = version * 100 (v1.71 = 00ABh)

07h spawn program (Back&Forth Professional only???)
ES:DI -> BF_SPAWN record (see #00232)
Return: AX = status
0000h if no task handles free
0001h spawn will occur when safe

08h get open tasks (documented only for Back&Forth Professional)
ES:DI -> task info buffer (see #00231,#00233)
Return: AX = number of open tasks (max 20)
Note: the supplied buffer must be large enough to hold 21 task entries

09h (reserved)
---Back&Forth Professional---

0Ah get active clipboard filename
Return: DX:AX -> ASCIZ clipboard filename

0Bh get active task number
AX = active task number (00h-13h)
BX = number of tasks allocated
DX = maximum number of tasks

0Ch (reserved)
0Dh (reserved)
0Eh (reserved)
0Fh stuff string into keyboard buffer
ES:DI -> ASCIZ string to be stuffed
Return: nothing

10h check if in graphics mode
Return: AX = state
0000h color text mode

```

    0004h mono text mode
    FFFFh graphics mode
    11h get Back&Forth Professional user number
Return: AX = user number (0000h-00FFh)
    12h switch task by task number
DX = task number
Return: AX = status
    0000h attempted to switch to active task
    0001h task switch will occur when safe
    FFFFh invalid task number
    13h delete (kill) task
DX = task number
Return: AX = status
    0000h attempted to delete the active task
    0001h successfully deleted
    FFFFh invalid task number
Note: the active task number will change if the deleted task
      was lower in the task list than the active task
    14h get next available task handle
Return: AX = next available task handle
    FFFFh if task table is full

```

Program: Back & Forth is a task switcher by Progressive Solutions, Inc.

SeeAlso: INT 11/AX=FFFEh

Index: installation check;BACK&FORTH

Format of Back&Forth task info buffer:

Offset Size Description (Table 00231)

```

    00h 21 BYTES ASCIZ task name
    15h BYTE hotkey shift state (as for INT 16/AH=02h)
    16h WORD hotkey scan code (see also #00006)
    18h WORD program ID

```

Index: hotkeys;Back&Forth

SeeAlso: #00232,#00233

Format of Back&Forth Professional BF_SPAWN record:

Offset Size Description (Table 00232)

```

    00h 21 BYTES task description
    15h BYTE flag: disable hotkeys
    16h WORD environment size in bytes
    18h BYTE hotkey shift flags
    19h WORD hotkey scancode

```

1Bh WORD maximum number of EMS pages
 1Dh WORD required memory in KBytes
 1Fh 3 BYTES DESQview-style two-letter program ID
 22h 13 BYTES base name of program to be run (no path or extension)
 2Fh 66 BYTES directory from which to start program
 71h 66 BYTES initial current directory for program
 SeeAlso: #00231,#00233

Format of Back&Forth Professional BF_TASK record:

Offset	Size	Description (Table 00233)
00h	DWORD	Unix-style task start time (seconds since 1970/1/1)
04h	21 BYTES	task description
19h	DWORD	elapsed time in task (seconds)
1Dh	WORD	task ID
1Fh	BYTE	task hotkey keyboard flags
20h	WORD	task hotkey scan code (see also #00006)
22h	DWORD	time task was suspended/exited
26h	WORD	task handle

SeeAlso: #00231,#00232

-----B-1300-----

INT 13 - DISK - RESET DISK SYSTEM

AH = 00h

DL = drive (if bit 7 is set both hard disks and floppy disks reset)

Return: AH = status (see #00234)

CF clear if successful (returned AH=00h)

CF set on error

Note: forces controller to recalibrate drive heads (seek to track 0)
 for PS/2 35SX, 35LS, 40SX and L40SX, as well as many other systems,
 both the master drive and the slave drive respond to the Reset
 function that is issued to either drive

SeeAlso: AH=0Dh,AH=11h,INT 21/AH=0Dh,INT 4D/AH=00h"TI Professional"

SeeAlso: INT 56"Tandy 2000",MEM 0040h:003Eh

-----B-1301-----

INT 13 - DISK - GET STATUS OF LAST OPERATION

AH = 01h

DL = drive (bit 7 set for hard disk)

Return: CF clear if successful (returned status 00h)

CF set on error

AH = status of previous operation (see #00234)

Note: some BIOSes return the status in AL; the PS/2 Model 30/286 returns the
 status in both AH and AL

SeeAlso: AH=00h,INT 4D/AH=01h,MEM 0040h:0041h,MEM 0040h:0074h

(Table 00234)

Values for disk operation status:

00h successful completion
01h invalid function in AH or invalid parameter
02h address mark not found
03h disk write-protected
04h sector not found/read error
05h reset failed (hard disk)
05h data did not verify correctly (TI Professional PC)
06h disk changed (floppy)
07h drive parameter activity failed (hard disk)
08h DMA overrun
09h data boundary error (attempted DMA across 64K boundary or >80h sectors)
0Ah bad sector detected (hard disk)
0Bh bad track detected (hard disk)
0Ch unsupported track or invalid media
0Dh invalid number of sectors on format (PS/2 hard disk)
0Eh control data address mark detected (hard disk)
0Fh DMA arbitration level out of range (hard disk)
10h uncorrectable CRC or ECC error on read
11h data ECC corrected (hard disk)
20h controller failure
31h no media in drive (IBM/MS INT 13 extensions)
32h incorrect drive type stored in CMOS (Compaq)
40h seek failed
80h timeout (not ready)
AAh drive not ready (hard disk)
B0h volume not locked in drive (INT 13 extensions)
B1h volume locked in drive (INT 13 extensions)
B2h volume not removable (INT 13 extensions)
B3h volume in use (INT 13 extensions)
B4h lock count exceeded (INT 13 extensions)
B5h valid eject request failed (INT 13 extensions)
B6h volume present but read protected (INT 13 extensions)
BBh undefined error (hard disk)
CCh write fault (hard disk)
E0h status register error (hard disk)
FFh sense operation failed (hard disk)

SeeAlso: #M0022

-----B-1302-----

INT 13 - DISK - READ SECTOR(S) INTO MEMORY

AH = 02h

AL = number of sectors to read (must be nonzero)

CH = low eight bits of cylinder number

CL = sector number 1-63 (bits 0-5)

high two bits of cylinder (bits 6-7, hard disk only)

DH = head number

DL = drive number (bit 7 set for hard disk)

ES:BX -> data buffer

Return: CF set on error

if AH = 11h (corrected ECC error), AL = burst length

CF clear if successful

AH = status (see #00234)

AL = number of sectors transferred (only valid if CF set for some BIOSes)

Notes: errors on a floppy may be due to the motor failing to spin up quickly enough; the read should be retried at least three times, resetting the disk with AH=00h between attempts

most BIOSes support "multitrack" reads, where the value in AL exceeds the number of sectors remaining on the track, in which case any additional sectors are read beginning at sector 1 on the following head in the same cylinder; the MSDOS CONFIG.SYS command MULTITRACK (or the Novell DOS DEBLOCK=) can be used to force DOS to split disk accesses which would wrap across a track boundary into two separate calls

the IBM AT BIOS and many other BIOSes use only the low four bits of DH (head number) since the WD-1003 controller which is the standard AT controller (and the controller that IDE emulates) only supports 16 heads

AWARD AT BIOS and AMI 386sx BIOS have been extended to handle more than 1024 cylinders by placing bits 10 and 11 of the cylinder number into bits 6 and 7 of DH

under Windows95, a volume must be locked (see INT 21/AX=440Dh/CX=084Bh) in order to perform direct accesses such as INT 13h reads and writes

all versions of MS-DOS (including MS-DOS 7 [Windows 95]) have a bug which prevents booting on hard disks with 256 heads (FFh), so many modern BIOSes provide mappings with at most 255 (FEh) heads

some cache drivers flush their buffers when detecting that DOS is bypassed by directly issuing INT 13h from applications. A dummy read can be used as one of several methods to force cache

flushing for unknown caches (e.g. before rebooting).

BUGS: When reading from floppies, some AMI BIOSes (around 1990-1991) trash the byte following the data buffer, if it is not arranged to an even memory boundary. A workaround is to either make the buffer word aligned (which may also help to speed up things), or to add a dummy byte after the buffer.

MS-DOS may leave interrupts disabled on return from this function.

Apparently some BIOSes or intercepting resident software have bugs that may destroy DX on return or not properly set the Carry flag.

At least some Microsoft software frames calls to this function with PUSH DX, STC, INT 13h, STI, POP DX.

on the original IBM AT BIOS (1984/01/10) this function does not disable interrupts for harddisks (DL >= 80h). On these machines the MS-DOS/PC DOS IO.SYS/IBMBIO.COM installs a special filter to bypass the buggy code in the ROM (see CALL F000h:211Eh)

SeeAlso: AH=03h,AH=0Ah,AH=06h"V10DISK.SYS",AH=21h"PS/1",AH=42h"IBM"

SeeAlso: INT 21/AX=440Dh/CX=084Bh,INT 4D/AH=02h

-----B-1303-----

INT 13 - DISK - WRITE DISK SECTOR(S)

AH = 03h

AL = number of sectors to write (must be nonzero)

CH = low eight bits of cylinder number

CL = sector number 1-63 (bits 0-5)

high two bits of cylinder (bits 6-7, hard disk only)

DH = head number

DL = drive number (bit 7 set for hard disk)

ES:BX -> data buffer

Return: CF set on error

CF clear if successful

AH = status (see #00234)

AL = number of sectors transferred

(only valid if CF set for some BIOSes)

Notes: errors on a floppy may be due to the motor failing to spin up quickly enough; the write should be retried at least three times, resetting the disk with AH=00h between attempts

most BIOSes support "multitrack" writes, where the value in AL exceeds the number of sectors remaining on the track, in which case any additional sectors are written beginning at sector 1 on the following head in the same cylinder; the CONFIG.SYS command MULTITRACK can be used to force DOS to split disk accesses which would wrap across a track boundary into two separate calls

the IBM AT BIOS and many other BIOSes use only the low four bits of DH (head number) since the WD-1003 controller which is the standard AT controller (and the controller that IDE emulates) only supports 16 heads

AWARD AT BIOS and AMI 386sx BIOS have been extended to handle more than 1024 cylinders by placing bits 10 and 11 of the cylinder number into bits 6 and 7 of DH

under Windows95, an application must issue a physical volume lock on the drive via INT 21/AX=440Dh before it can successfully write to the disk with this function

SeeAlso: AH=02h,AH=0Bh,AH=07h"V10DISK.SYS",AH=22h"PS/1",AH=43h"IBM"

SeeAlso: INT 21/AX=440Dh"DOS 3.2+",INT 4D/AH=03h

-----B-1304-----

INT 13 - DISK - VERIFY DISK SECTOR(S)

AH = 04h

AL = number of sectors to verify (must be nonzero)

CH = low eight bits of cylinder number

CL = sector number 1-63 (bits 0-5)

high two bits of cylinder (bits 6-7, hard disk only)

DH = head number

DL = drive number (bit 7 set for hard disk)

ES:BX -> data buffer (PC,XT,AT with BIOS prior to 1985/11/15)

Return: CF set on error

CF clear if successful

AH = status (see #00234)

AL = number of sectors verified

Notes: errors on a floppy may be due to the motor failing to spin up quickly enough (timeout error 80h); the write should be retried at least three times, resetting the disk with AH=00h between attempts on floppys, the operation should also be retried on media change (06h) detection.

this function does not compare the disk with memory, it merely checks whether the sector's stored CRC matches the data's actual CRC

the IBM AT BIOS and many other BIOSes use only the low four bits of DH (head number) since the WD-1003 controller which is the standard AT controller (and the controller that IDE emulates) only supports 16 heads

AWARD AT BIOS and AMI 386sx BIOS have been extended to handle more than 1024 cylinders by placing bits 10 and 11 of the cylinder number into bits 6 and 7 of DH

BUG: some Epson ROM BIOSes sometimes have problems properly handling this

function. The workaround is to reset the disk (INT 13/AH=00h) before
the call.

SeeAlso: AH=02h,AH=44h,INT 4D/AH=04h,INT 4D/AH=06h

-----B-1305-----

INT 13 - FLOPPY - FORMAT TRACK

AH = 05h

AL = number of sectors to format

CH = track number

DH = head number

DL = drive number

ES:BX -> address field buffer (see #00235)

Return: CF set on error

CF clear if successful

AH = status (see #00234)

Notes: on AT or higher, call AH=17h first

the number of sectors per track is read from the diskette parameter
table pointed at by INT 1E

BUG: some old Compaq BIOSes have a bug when attempting to call this function

from Windows Standard Mode with EMM386 loaded. A possible workaround

is to call this function from Real Mode e.g. through DPMI function

"Call Real Mode function with FAR return" (see INT 31/AX=0301h).

SeeAlso: AH=05h"FIXED",AH=17h,AH=18h,INT 1E

Format of floppy format address field buffer entry (one per sector in track):

Offset Size Description (Table 00235)

00h BYTE track number

01h BYTE head number (0-based)

02h BYTE sector number

03h BYTE sector size (00h=128 bytes, 01h=256 bytes, 02h=512, 03h=1024)

-----B-1305-----

INT 13 - FIXED DISK - FORMAT TRACK

AH = 05h

AL = interleave value (XT-type controllers only)

ES:BX -> 512-byte format buffer

the first 2*(sectors/track) bytes contain F,N for each sector

F = sector type

00h for good sector

20h to unassign from alternate location

40h to assign to alternate location

80h for bad sector

N = sector number

CH = cylinder number (bits 8,9 in high bits of CL)
CL = high bits of cylinder number (bits 7,6)
DH = head
DL = drive

Return: CF set on error
CF clear if successful

AH = status code (see #00234)

Notes: AWARD AT BIOS and AMI 386sx BIOS have been extended to handle more than 1024 cylinders by placing bits 10 and 11 of the cylinder number into bits 6 and 7 of DH

for XT-type controllers on an AT or higher, AH=0Fh should be called first

the IBM AT BIOS and many other BIOSes use only the low four bits of DH (head number) since the WD-1003 controller which is the standard AT controller (and the controller that IDE emulates) only supports 16 heads

not all controller support sector types 20h and 40h

under Windows95, an application must issue a physical volume lock on the drive via INT 21/AX=440Dh before it can successfully write to the disk with this function

SeeAlso: AH=05h"FLOPPY",AH=06h"FIXED",AH=07h"FIXED",AH=0Fh,AH=18h,AH=1Ah

-----d-1305-----

INT 13 - Future Domain SCSI BIOS - SEND SCSI MODE SELECT COMMAND

AH = 05h

DL = hard drive ID

ES:BX -> mode select data (see #00236)

Return: CF set on error

CF clear if successful

AH = status code (see #00234)

Notes: this function can be called before AH=07h"SCSI" or AH=06h"SCSI" to format a SCSI disk with the desired parameters

the mode select data below is from the SCSI-1 specification

the TMC-950 does not support any Future Domain BIOS calls; instead, it provides a full CAM implementation (see INT 4F/AX=8100h)

SeeAlso: AH=06h"SCSI",AH=07h"SCSI",INT 4F/AX=8100h

Format of Future Domain SCSI mode select data:

Offset Size Description (Table 00236)

00h BYTE number of bytes of remaining data (12 + vendor unique length)

01h BYTE reserved (0)

02h BYTE medium type (0 for hard disk)

03h BYTE reserved (0)
04h BYTE block descriptor length (8)
05h BYTE density code (0 for hard disk)
06h 3 BYTES (big-endian) number of blocks (000000h for entire disk)
09h BYTE reserved (0)
0Ah 3 BYTES (big-endian) block length (512 standard, or 256)
0Dh ??? vendor-specific parameter bytes (optional)

-----d-13057FSI324D-----

INT 13 - 2M - FORMAT TRACK

AX = 057Fh

SI = 324Dh ("2M")

CH = track number

DH = head number

DL = drive number

ES:BX -> boot sector of future 2M diskette

Return: CF set on error

CF clear if successful

AH = status (see #00234)

Program: 2M is a TSR developed by Ciriaco Garc de Celis to support
non standard diskettes with 820-902/1476-1558K (5.25 DD/HD)
and 984-1066/1804-1886K/3608-3772K (3.5 DD/HD/ED)

InstallCheck: must search for a "CiriSOFT:2M:1.3" or "CiriSOFT:2MX:3.0" or
similar (recomended ":2M:", ":2MX:", or ":2MB:" substrings) in the
CiriSOFT TSR interface

Notes: it is not necessary to call AH=17h or AH=18h first (will be ignored)

the diskette format must always begin on cylinder 0 head 0

the boot sector can be obtained from an already-formatted 2M diskette
(by calling AH=02h with head number 00h in 2M v1.x and with head
number 80h for 2M v2+)

since 2M v2.0, the BOOT sector is emulated using the first physical
sector of FAT2; the second-sixth physical sectors of FAT2 in HD or ED
diskettes store the SuperBOOT code. To skip the FAT2 emulation (using
FAT1) of 2M, in order to read the SuperBOOT code, head number must be
80h-81h instead 0-1 (bit 7 active) in standard read/write functions.
This lets diskcopy programs format 2M target diskettes copying
SuperBOOT code. If the target diskette is already 2MF formatted
(containing boot code) this trick it is not necessary.

when using STV technology (offset 65 of boot sector equal to 1) it is
necessary to write the full track before formatting (except track 0
side 0) to complete the format and skip future CRC errors on read; in
track 0 side 1 the head used must be 81h instead 1. Diskcopy programs

may do a format-write-verify sequential phases to improve performance

SeeAlso: AH=05h"FLOPPY",AH=18h/CX=5055h,INT 2F"CiriSOFT"

-----B-1306-----

INT 13 - FIXED DISK - FORMAT TRACK AND SET BAD SECTOR FLAGS (XT,PORT)

AH = 06h

AL = interleave value

CH = cylinder number (bits 8,9 in high bits of CL)

CL = sector number

DH = head

DL = drive

Return: AH = status code (see #00234)

Note: AWARD AT BIOS and AMI 386sx BIOS have been extended to handle more than 1024 cylinders by placing bits 10 and 11 of the cylinder number into bits 6 and 7 of DH

SeeAlso: AH=05h"FIXED",AH=07h"FIXED"

-----d-1306-----

INT 13 - Future Domain SCSI BIOS - FORMAT DRIVE WITH BAD SECTOR MAPPING

AH = 06h

AL = interleave

(0 = default, 1 = consecutive sectors, 2 - 255 = vendor unique)

DL = hard drive ID

DH = defect list info (see #00237)

ES:BX -> defect table A, B or C (see #00238,#00239,#00240)

Return: CF set on error

CF clear if successful

AH = status code (see #00234)

Notes: block addresses must be in ascending order (for table B, cylinder is

most significant, byte from index least significant; for table C,

cylinder is most significant, sector number least significant)

table B defect bytes from index of FFFFFFFFh indicates that the entire track shall be reassigned

table C defect sector number of FFFFFFFFh indicates that the entire track shall be reassigned

the TMC-950 does not support any Future Domain BIOS calls; instead, it provides a full CAM implementation (see INT 4F/AX=8100h)

SeeAlso: AH=05h"SCSI",AH=06h"FIXED",AH=07h"SCSI"

Bitfields for Future Domain SCSI defect list info:

Bit(s) Description (Table 00237)

7-5 drive LUN

4 defect list is available

```
3 defect list is complete (erase drive's defect list)
2-0 defect table format
  (000=use defect table A, 100=use defect table B,
   101=use defect table C)
```

Format of Future Domain SCSI defect table A:

```
Offset Size Description (Table 00238)
00h WORD number of bytes remaining in table
02h BYTE reserved (0)
03h BYTE reserved (0)
04h WORD (big-endian) defect list length (4*number of defects)
06h 4 DWORDs (big-endian) defect block addresses
```

Format of Future Domain SCSI defect table B:

```
Offset Size Description (Table 00239)
00h WORD number of bytes remaining in table
02h BYTE reserved (0)
03h BYTE reserved (0)
04h WORD (big-endian) defect list length (8*number of defects)
06h 8N BYTES defect list [array] (see #00241)
```

Format of Future Domain SCSI defect table C:

```
Offset Size Description (Table 00240)
00h WORD number of bytes remaining in table
02h BYTE reserved (0)
03h BYTE reserved (0)
04h WORD (big-endian) defect list length (8*number of defects)
06h 8N BYTES defect list [array] (see #00241)
```

Format of Future Domain SCSI defect list entry:

```
Offset Size Description (Table 00241)
00h 3 BYTES (big-endian) cylinder number of defect
03h BYTE head number of defect
04h DWORD (big-endian) defect bytes from index
```

-----d-1306-----

INT 13 - Adaptec AHA-154xA/Bustek BT-542 BIOS - IDENTIFY SCSI DEVICES

AH = 06h

Return: AH = status code (see #00234)

CF clear if successful

AL = first drive supported

(80h nonconcurrent operation, 81h concurrent operation)

CF set on error
Desc: determine the number of the first supported SCSI drive
Note: the return value is 80h when two SCSI drives are supported, 81h if
only one SCSI drive is installed
SeeAlso: AH=08h"PC",#00732 at INT 1A/AX=B102h
-----d-1306-----
INT 13 - V10DISK.SYS - READ DELETED SECTORS
AH = 06h
AL = number of sectors
CH = cylinder number (bits 8,9 in high bits of CL)
CL = sector number
DH = head
DL = drive
ES:BX -> buffer
Return: AH = status code (see #00234)
Program: V10DISK.SYS is a driver for the Flagstaff Engineering 8" floppies
SeeAlso: AH=02h,AH=07h"V10DISK.SYS"
-----B-1307-----
INT 13 - FIXED DISK - FORMAT DRIVE STARTING AT GIVEN TRACK (XT,PORT)
AH = 07h
AL = interleave value (XT only)
ES:BX = 512-byte format buffer (see AH=05h)
CH = cylinder number (bits 8,9 in high bits of CL)
CL = sector number
DH = head
DL = drive
Return: AH = status code (see #00234)
Note: AWARD AT BIOS and AMI 386sx BIOS have been extended to handle more
than 1024 cylinders by placing bits 10 and 11 of the cylinder number
into bits 6 and 7 of DH
SeeAlso: AH=05h"FIXED",AH=06h"FIXED",AH=1Ah
-----d-1307-----
INT 13 - Future Domain SCSI BIOS - FORMAT DRIVE
AH = 07h
AL = interleave (0 = default, 1 = consecutive sectors,
2 - 255 = vendor unique)
DL = hard drive ID
Return: CF set on error
CF clear if successful
AH = status code (see #00234)
SeeAlso: AH=05h"SCSI",AH=06h"SCSI",AH=07h"FIXED"

-----d-1307-----

INT 13 - V10DISK.SYS - WRITE DELETED SECTORS

AH = 07h
AL = number of sectors
CH = cylinder number (bits 8,9 in high bits of CL)
CL = sector number
DH = head
DL = drive
ES:BX -> buffer

Return: AH = status code (see #00234)

Program: V10DISK.SYS is a driver for the Flagstaff Engineering 8" floppies

SeeAlso: AH=03h,AH=06h"V10DISK.SYS"

-----B-1308-----

INT 13 - DISK - GET DRIVE PARAMETERS (PC,XT286,CONV,PS,ESDI,SCSI)

AH = 08h
DL = drive (bit 7 set for hard disk)
ES:DI = 0000h:0000h to guard against BIOS bugs

Return: CF set on error

AH = status (07h) (see #00234)
CF clear if successful
AH = 00h
AL = 00h on at least some BIOSes
BL = drive type (AT/PS2 floppies only) (see #00242)
CH = low eight bits of maximum cylinder number
CL = maximum sector number (bits 5-0)
high two bits of maximum cylinder number (bits 7-6)
DH = maximum head number
DL = number of drives
ES:DI -> drive parameter table (floppies only)

Notes: may return successful even though specified drive is greater than the number of attached drives of that type (floppy/hard); check DL to ensure validity

for systems predating the IBM AT, this call is only valid for hard disks, as it is implemented by the hard disk BIOS rather than the ROM BIOS

the IBM ROM-BIOS returns the total number of hard disks attached to the system regardless of whether DL >= 80h on entry.

Toshiba laptops with HardRAM return DL=02h when called with DL=80h, but fail on DL=81h. The BIOS data at 40h:75h correctly reports 01h. may indicate only two drives present even if more are attached; to ensure a correct count, one can use AH=15h to scan through possible

drives

Reportedly some Compaq BIOSes with more than one hard disk controller return only the number of drives DL attached to the corresponding controller as specified by the DL value on entry. However, on Compaq machines with "COMPAQ" signature at F000h:FFEAh, MS-DOS/PC DOS IO.SYS/IBMBIO.COM call INT 15/AX=E400h and INT 15/AX=E480h to enable Compaq "mode 2" before retrieving the count of hard disks installed in the system (DL) from this function.

the maximum cylinder number reported in CX is usually two less than the total cylinder count reported in the fixed disk parameter table (see INT 41h,INT 46h) because early hard disks used the last cylinder for testing purposes; however, on some Zenith machines, the maximum cylinder number reportedly is three less than the count in the fixed disk parameter table.

for BIOSes which reserve the last cylinder for testing purposes, the cylinder count is automatically decremented

on PS/1s with IBM ROM DOS 4, nonexistent drives return CF clear, BX=CX=0000h, and ES:DI = 0000h:0000h

machines with lost CMOS memory may return invalid data for floppy drives. In this situation CF is cleared, but AX,BX,CX,DX,DH,DI, and ES contain only 0. At least under some circumstances, MS-DOS/PC DOS IO.SYS/IBMBIO.COM just assumes a 360 KB floppy if it sees CH to be zero for a floppy.

the PC-Tools PCFORMAT program requires that AL=00h before it will proceed with the formatting

if this function fails, an alternative way to retrieve the number of floppy drives installed in the system is to call INT 11h.

In fact, the MS-DOS/PC-DOS IO.SYS/IBMBIO.COM attempts to get the number of floppy drives installed from INT 13/AH=08h, when INT 11h AX bit 0 indicates there are no floppy drives installed. In addition to testing the CF flag, it only trusts the result when the number of sectors (CL preset to zero) is non-zero after the call.

BUGS: several different Compaq BIOSes incorrectly report high-numbered drives (such as 90h, B0h, D0h, and F0h) as present, giving them the same geometry as drive 80h; as a workaround, scan through disk numbers, stopping as soon as the number of valid drives encountered equals the value in 0040h:0075h

a bug in Leading Edge 8088 BIOS 3.10 causes the DI,SI,BP,DS, and ES registers to be destroyed

some Toshiba BIOSes (at least before 1995, maybe some laptops??? with 1.44 MB floppies) have a bug where they do not set the ES:DI

vector even for floppy drives. Hence these registers should be preset with zero before the call and checked to be non-zero on return before using them. Also it seems these BIOSes can return wrong info in BL and CX, as S/DOS 1.0 can be configured to preset these registers as for an 1.44 MB floppy.

the PS/2 Model 30 fails to reset the bus after INT 13/AH=08h and INT 13/AH=15h. A workaround is to monitor for these functions and perform a transparent INT 13/AH=01h status read afterwards. This will reset the bus. The MS-DOS 6.0 IO.SYS takes care of this by installing a special INT 13h interceptor for this purpose. AD-DOS may leave interrupts disabled on return from this function.

Some Microsoft software explicitly sets STI after return.

SeeAlso: AH=06h"Adaptec",AH=13h"SyQuest",AH=48h,AH=15h,INT 1E

SeeAlso: INT 41"HARD DISK 0"

(Table 00242)

Values for diskette drive type:

01h 360K
02h 1.2M
03h 720K
04h 1.44M
05h ??? (reportedly an obscure drive type shipped on some IBM machines)
2.88M on some machines (at least AMI 486 BIOS)
06h 2.88M
10h ATAPI Removable Media Device

-----d-1308-----

INT 13 - V10DISK.SYS - SET FORMAT

AH = 08h
AL = number of sectors
CH = cylinder number (bits 8,9 in high bits of CL)
CL = sector number
DH = head
DL = drive

Return: AH = status code (see #00234)

Program: V10DISK.SYS is a driver for the Flagstaff Engineering 8" floppies

Note: details not available

SeeAlso: AH=03h,AH=06h"V10DISK.SYS"

-----y-130800DLF0-----

INT 13 - SecureDrive - INSTALLATION CHECK

AX = 08000h
DL = F0h

Return: AX = EDCBh for version 1.0-1.2

AX = EDCCh for version 1.3

CX = code segment

DX = data address within code segment

Program: SecureDrive by Mike Ingle <mikeingle@delphi.com> allows you to create an encrypted partition on your harddisk.

-----B-1309-----

INT 13 - HARD DISK - INITIALIZE CONTROLLER WITH DRIVE PARAMETERS (AT,PS)

AH = 09h

DL = drive (80h for first, 81h for second)

Return: CF clear if successful

CF set on error

AH = status (see #00234)

Notes: on the PC and XT, this function uses the parameter table pointed at by

INT 41

on the AT and later, this function uses the parameter table pointed at by INT 41 if DL=80h, and the parameter table pointed at by INT 46 if DL=81h

SeeAlso: INT 41"HARD DISK 0",INT 46"HARD DISK 1"

-----B-130A-----

INT 13 - HARD DISK - READ LONG SECTOR(S) (AT and later)

AH = 0Ah

AL = number of sectors (01h may be only value supported)

CH = low eight bits of cylinder number

CL = sector number (bits 5-0)

high two bits of cylinder number (bits 7-6)

DH = head number

DL = drive number (80h = first, 81h = second)

ES:BX -> data buffer

Return: CF clear if successful

CF set on error

AH = status (see #00234)

AL = number of sectors transferred

Notes: this function reads in four to seven bytes of error-correcting code

along with each sector's worth of information

data errors are not automatically corrected, and the read is aborted after the first sector with an ECC error

used for diagnostics only on PS/2 systems; IBM officially classifies this function as optional

BUG: on the original IBM AT BIOS (1984/01/10) this function does not disable interrupts for harddisks (DL >= 80h). On these machines the MS-DOS/

PC DOS IO.SYS/IBMBIO.COM installs a special filter to bypass the

buggy code in the ROM (see CALL F000h:211Eh)

SeeAlso: AH=02h,AH=0Bh,MEM 0040h:0074h

-----B-130B-----

INT 13 - HARD DISK - WRITE LONG SECTOR(S) (AT and later)

AH = 0Bh

AL = number of sectors (01h may be only value supported)

CH = low eight bits of cylinder number

CL = sector number (bits 5-0)

high two bits of cylinder number (bits 7-6)

DH = head number

DL = drive number (80h = first, 81h = second)

ES:BX -> data buffer

Return: CF clear if successful

CF set on error

AH = status (see #00234)

AL = number of sectors transferred

Notes: each sector's worth of data must be followed by four to seven bytes of
error-correction information

used for diagnostics only on PS/2 systems; IBM officially classifies
this function as optional

SeeAlso: AH=03h,AH=0Ah,MEM 0040h:0074h

-----B-130C-----

INT 13 - HARD DISK - SEEK TO CYLINDER

AH = 0Ch

CH = low eight bits of cylinder number

CL = sector number (bits 5-0)

high two bits of cylinder number (bits 7-6)

DH = head number

DL = drive number (80h = first, 81h = second hard disk)

Return: CF set on error

CF clear if successful

AH = status (see #00234)

SeeAlso: AH=00h,AH=02h,AH=0Ah,AH=47h

-----B-130D-----

INT 13 - HARD DISK - RESET HARD DISKS

AH = 0Dh

DL = drive number (80h = first, 81h = second hard disk)

Return: CF set on error

CF clear if successful

AH = status (see #00234)

Notes: reinitializes the hard disk controller, resets the specified drive's parameters, and recalibrates the drive's heads (seek to track 0) for PS/2 35SX, 35LS, 40SX and L40SX, as well as many other systems, both the master drive and the slave drive respond to the Reset function that is issued to either drive not for PS/2 ESDI drives

SeeAlso: AH=00h,INT 21/AH=0Dh

-----B-130E-----

INT 13 - HARD DISK - READ SECTOR BUFFER (XT only)

AH = 0Eh

DL = drive number (80h = first, 81h = second hard disk)

ES:BX -> buffer

Return: CF set on error

CF clear if successful

AH = status code (see #00234)

Notes: transfers controller's sector buffer. No data is read from the drive used for diagnostics only on PS/2 systems

SeeAlso: AH=0Ah

-----B-130F-----

INT 13 - HARD DISK - WRITE SECTOR BUFFER (XT only)

AH = 0Fh

DL = drive number (80h = first, 81h = second hard disk)

ES:BX -> buffer

Return: CF set on error

CF clear if successful

AH = status code (see #00234)

Notes: does not write data to the drive should be called before formatting to initialize an XT-type controller's sector buffer used for diagnostics only on PS/2 systems

SeeAlso: AH=0Bh

-----B-1310-----

INT 13 - HARD DISK - CHECK IF DRIVE READY

AH = 10h

DL = drive number (80h = first, 81h = second hard disk)

Return: CF set on error

CF clear if successful

AH = status (see #00234 at AH=01h)

SeeAlso: AH=11h

-----B-1311-----

INT 13 - HARD DISK - RECALIBRATE DRIVE

AH = 11h
DL = drive number (80h = first, 81h = second hard disk)
Return: CF set on error
CF clear if successful
AH = status (see #00234 at AH=01h)
Note: causes hard disk controller to seek the specified drive to cylinder 0
SeeAlso: AH=00h,AH=0Ch,AH=10h,AH=19h"FIXED DISK",MEM 0040h:003Eh

-----B-1312-----

INT 13 - HARD DISK - CONTROLLER RAM DIAGNOSTIC (XT,PS)
AH = 12h
DL = drive number (80h = first, 81h = second hard disk)
Return: CF set on error
CF clear if successful
AH = status code (see #00234 at AH=01h)
AL = 00h

SeeAlso: AH=13h,AH=14h

-----d-1312-----

INT 13 - Future Domain SCSI CONTROLLER - STOP SCSI DISK
AH = 12h
DL = hard drive ID
Return: CF set on error
CF clear if successful
AH = status code (see #00234 at AH=01h)
Notes: available at least on the TMC-870 8-bit SCSI controller BIOS v6.0A
if the given drive is a SCSI device, the SCSI Stop Unit command is sent
and either "Disk prepared for shipping" or "Disk Stop command failed"
is displayed
the TMC-950 does not support any Future Domain BIOS calls; instead,
it provides a full CAM implementation (see INT 4F/AX=8100h)

-----d-1312-----

INT 13 - SyQuest - START/STOP SCSI DISK
AH = 12h
AL = subfunction
00h start disk
01h stop disk
CX = wait flag
00h wait for ready
01h don't wait for ready
DL = hard drive ID (bit 7 for hard disks must be set)
Return: CF set on error
CF clear if successful

AH = status
00h successful
01h invalid function request
80h timeout

SeeAlso: AH=12h"Future Domain",AH=13h"SyQuest"

-----B-1313-----

INT 13 - HARD DISK - DRIVE DIAGNOSTIC (XT,PS)

AH = 13h
DL = drive number (80h = first, 81h = second hard disk)

Return: CF set on error

CF clear if successful
AH = status code (see #00234 at AH=01h)
AL = 00h

SeeAlso: AH=12h"HARD DISK",AH=14h"HARD DISK"

-----d-1313-----

INT 13 - SyQuest - READ DRIVE PARAMATERS (for DOS 5+)

AH = 13h
DL = drive ID (bit 7 set for hard disks)

Return: CF set on error

AH = status (07h) (see #00234 at AH=01h)
CF clear if successful
AH = 00h
BL = drive type (AT/PS2 floppies only) (see #00242)
CH = low eight bits of maximum cylinder number
CL = maximum sector number (bits 5-0)
high two bits of maximum cylinder number (bits 7-6)
DH = maximum head number
DL = number of drives
ES:DI -> drive parameter table (floppies only)

Notes: the return values are identical to the standard INT 13/AH=08h, but the number of drives is not limited to 2, so scanning all possible drive numbers with the Read DASD Type call (AH=15h) should generally be preferred to determine the number of drives attached to the system.

SeeAlso: AH=08h"PC",AH=12h"SyQuest",AH=15h,AH=59h"SyQuest"

-----B-1314-----

INT 13 - HARD DISK - CONTROLLER INTERNAL DIAGNOSTIC

AH = 14h
Return: CF set on error
CF clear if successful
AH = status code (see #00234 at AH=01h)

AL = 00h

SeeAlso: AH=12h,AH=13h

-----B-1315-----

INT 13 - DISK - GET DISK TYPE (XT 1986/1/10 or later,XT286,AT,PS)

AH = 15h

DL = drive number (bit 7 set for hard disk)

(AL = FFh, CX = FFFFh, see Note)

Return: CF clear if successful

AH = type code

00h no such drive

(SpeedStor) AL = 03h hard disk

CX:DX = number of 512-byte sectors

01h floppy without change-line support

02h floppy (or other removable drive) with change-line support

03h hard disk

CX:DX = number of 512-byte sectors

CF set on error

AH = status (see #00234 at AH=01h)

Note: SyQuest can report type 01h or 02h for 'hard disks', since its media is removable

BUGS: many versions of the Award 486 BIOS do not return the sector count

because the BIOS exit code restores CX and DX to their original values after the function had already set them to correct values

Some releases of PC Tools REBUILD preset CX=FFFFh and only trust the results if CH <= 2 on return (which would cut off drives > 16 Gb).

several different Compaq BIOSes incorrectly report high-numbered drives (such as 90h, B0h, D0h, and F0h) as present, giving them the same geometry as drive 80h; as a workaround, scan through disk numbers, stopping as soon as the number of valid drives encountered equals the value in 0040h:0075h

the PS/2 Model 30 fails to reset the bus after INT 13/AH=08h and

INT 13/AH=15h. A workaround is to monitor for these functions

and perform a transparent INT 13/AH=01h status read afterwards.

This will reset the bus. The MS-DOS 6.0 IO.SYS takes care of

this by installing a special INT 13h interceptor for this purpose.

Some releases of SpeedStor have a bug where it reports AX=0003h instead

of correctly reporting AH=03h for hard disks. A possible workaround

when testing for hard disks is to check for AH=03h and AX=0003h.

In this case this function should be invoked with a bogus fixed

value in AL, e.g. AL=FFh.

SeeAlso: AH=08h,AH=16h,AH=17h,AH=19h"SCSI",MEM 0040h:0075h

-----B-1316-----

INT 13 - FLOPPY DISK - DETECT DISK CHANGE (XT 1986/1/10 or later,XT286,AT,PS)

AH = 16h

DL = drive number (00h-7Fh)

SI = 0000h (to avoid crash on AT&T 6300)

Return: CF clear if change line inactive

AH = 00h (disk not changed)

CF set if change line active

AH = status

01h invalid command (SyQuest)

06h change line active or not supported

80h drive not ready or not present

Notes: call AH=15h first to determine whether the drive supports a change line

this call also clears the media-change status, so that a disk change is only reported once

BUGS: some versions of Award 386 Modular BIOS and AMI BIOS fail to clear the media-change status

AT&T 6300 WGS systems crash if SI <> 0 on entry.

some pre 1986/08/04 Compaq ROM BIOS have a serious bug where this function may re-configure a hard disk depending on what is located at ES:[BX] and data indexed to by it. MS-DOS/PC DOS IO.SYS/IBMBIO.COM install a special filter when they detect Compaq ROM BIOSes with earlier dates.

some Compaq 286 systems have a bug in all INT 13h functions >= 16h, which causes the byte at DS:0074h to be destroyed when called for hard disks (DL >= 80h). MS-DOS/PC DOS IO.SYS/IBMBIO.COM performs a test on this bug using this sub-function, and if found installs a special filter which points DS into ROM, so that it cannot cause any harm.

some drives (or controllers???) forget the change line status if another drive is accessed afterwards. The DOS BIOS takes care of this by not relying on the reported change line status when the change line is not active and a different drive is accessed, instead it reports "don't know" to the DOS kernel.

SeeAlso: AH=15h,AH=49h

-----B-1317-----

INT 13 - FLOPPY DISK - SET DISK TYPE FOR FORMAT (AT,PS)

AH = 17h

AL = format type

01h = 320/360K disk in 360K drive

02h = 320/360K disk in 1.2M drive
 03h = 1.2M disk in 1.2M drive
 04h = 720K disk in 720K or 1.44M drive

DL = drive number

Return: CF set on error

CF clear if successful

AH = status (see #00234 at AH=01h)

Note: this function does not handle 1.44M drives; use AH=18h instead

SeeAlso: AH=15h,AH=18h

-----d-131700-----

INT 13 - Future Domain SCSI CONTROLLER - GET INQUIRY INFO FROM SCSI DEVICE

AX = 1700h

CL = length of buffer

DL = hard drive ID

ES:BX -> buffer for info (see #00243)

Return: CF clear if successful

CH = number of bytes returned in buffer???

CF set on error

AH = status code (see #00234 at AH=01h)

Notes: this function is not available with 8-bit controller ROM versions < 7.0

information block bytes 5-n are vendor-specific in older SCSI devices

the TMC-950 does not support any Future Domain BIOS calls; instead,

it provides a full CAM implementation (see INT 4F/AX=8100h)

SeeAlso: AH=18h"SCSI",AH=1Bh"SCSI"

Format of Future Domain SCSI inquiry information block:

Offset Size Description (Table 00243)

00h BYTE device type

bits 0-4: peripheral device type (see #00244)

bits 5-7: peripheral qualifier (see #00245)

01h BYTE device type modifier

bits 0-6: device type modifier

bit 7: removable medium

02h BYTE SCSI version (see #00246)

03h BYTE data format/capabilities (see #00247)

04h BYTE additional data length (total remaining bytes)

05h 2 BYTES reserved

07h BYTE device capabilities (see #00248)

08h 8 BYTES vendor identification (space-padded ASCII)

10h 8 BYTES product identification (space-padded ASCII)

20h 4 BYTES product revision level (space-padded ASCII)

24h 20 BYTEs vendor specific
38h 40 BYTEs reserved
60h var vendor specific parameters

(Table 00244)

Values for Future Domain SCSI peripheral device type:

00h direct-access device (e.g., magnetic disk)
01h sequential-access device (e.g., magnetic tape)
02h printer device
03h processor device
04h write-once device (e.g., some optical disks)
05h CD-ROM device
06h scanner device
07h optical memory device (e.g., some optical disks)
08h medium changer device (e.g., jukeboxes)
09h communications device
0Ah (defined by ASC IT8)
0Bh (defined by ASC IT8)
0Ch-1Eh reserved
1Fh unknown or no device type

(Table 00245)

Values for Future Domain SCSI peripheral qualifier:

000b device is currently connected to this logical unit and available
001b target is capable of supporting the specified peripheral, but the
 physical device is not currently connected to this logical unit
010b reserved
011b target can't support a physical device on this logical unit
1xxb vendor specific

Bitfields for Future Domain SCSI version:

Bit(s) Description (Table 00246)

0-2 ANSI-approved version
 000 device might or might not comply to ANSI standard
 001 device complies to ANSI SCSI-1
 010 device complies to ANSI SCSI-2
 other reserved
3-5 ECMA version
6-7 ISO version

Bitfields for Future Domain SCSI data format/capabilities:

Bit(s) Description (Table 00247)
0-2 response data format
 000 information block is as specified in SCSI-1
 001 information block is as specified in CCS
 010 information block is as specified in SCSI-2
 other reserved
4-5 reserved
6 terminate I/O process supported
7 asynchronous event notification supported

Bitfields for Future Domain SCSI device capabilities:

Bit(s) Description (Table 00248)
0 device responds to RESET with a hard RESET
1 tagged command queuing supported
2 reserved
3 linked commands supported
4 synchronous data transfer supported
5 16-transfers supported
6 32-transfers supported
7 relative addressing supported

-----B-1318-----

INT 13 - DISK - SET MEDIA TYPE FOR FORMAT (AT model 3x9,XT2,XT286,PS)
 AH = 18h
 DL = drive number
 CH = lower 8 bits of highest cylinder number (number of cylinders - 1)
 CL = sectors per track (bits 0-5)
 top 2 bits of highest cylinder number (bits 6,7)

Return: AH = status
 00h requested combination supported
 01h function not available
 0Ch not supported or drive type unknown
 80h there is no disk in the drive

 ES:DI -> 11-byte parameter table (see #01264 at INT 1E)

Note: this function does not set the INT 1E vector to point at the returned
 parameter table; it is the caller's responsibility to do so

SeeAlso: AH=05h,AH=07h,AH=17h,INT 1E

-----d-1318-----

INT 13 - Future Domain SCSI BIOS - GET SCSI CONTROLLER INFORMATION
 AH = 18h
 DL = hard drive ID

Return: CF set on error

AH = status code (see #00234 at AH=01h)
 CF clear if successful
 AX = 4321h (magic number)
 CX = controller family code (see #00249)
 ---if family code=0200h
 DH = number of exclusively ROM-controlled SCSI devices
 DL = canonical SCSI device number for specified drive
 ---if family code <> 0200h
 BH = number of exclusively ROM-controlled SCSI devices
 BL = canonical SCSI device number for specified drive

Notes: also sets an internal flag (non-resettable) which prevents some controller messages from being displayed, allows writes to removable devices (use caution!), and enables the INT 13 interface for more than one drive (i.e. DL >= 81h) in at least some ROM versions

the TMC-950 does not support any Future Domain BIOS calls; instead, it provides a full CAM implementation (see INT 4F/AX=8100h)

SeeAlso: AH=05h"SCSI",AX=1700h"SCSI",AH=1Bh"SCSI",INT 4F/AX=8100h

(Table 00249)

Values for Future Domain SCSI controller family code:

0200h TMC-1680/? (ROM 3.0)
 0203h TMC-1650/1660/1670/1680 (ROM 2.0)
 040Ah TMC-820/830/840/850/860/870/875/880/885 (ROM <= 6.0A)
 050Dh TMC-840/841/880/881 (ROM 5.2D)
 0700h TMC-830/850/860/875/885 (ROM 7.0)

-----d-1318--CX5055-----

INT 13 - PU_1700.COM - INSTALLATION CHECK

AH = 18h
 CX = 5055h ('PU')
 DL = 00h

Return: AX = 7570h ('up') if PU_1700 is installed

Program: PU_1700 is a BIOS enhancer from PU Service Systems which permits formatting diskettes at higher capacity (1.78M instead of 1.44M)

SeeAlso: AX=057Fh/SI=324Dh"2M"

-----d-1318--CXD2C9-----

INT 13 - XDF.COM - API

AH = 18h
 CX = D2C9h ("R"+80h, "I"+80h = Roger Ivey)
 DX = 0000h
 BX = function

0000h installation check

Return: AH = 0Ch

CX = 7269h ("ri" = Roger Ivey)

ES = segment of driver

CF set

2F64h ("/d") disable the driver

Return: AH = 0Ch

ES:BX = pointer to activation flag (it is set to 0:
set it to 1 to enable the driver again)

CX = 7269h

CF set

2F75h ("/u") unload the driver (restore interrupts & free memory)

Return: AH = 0Ch

DL = 55h ("U") if successful
= 00h if fails

CX = 7269h

ES = segment of driver

CF set

AL, BX, DH, and DI destroyed

Program: XDF is a TSR provided with PC-DOS 7.0 to support XDF 1.84M disks,
developed by Roger D. Ivey

Note: After disabling or enabling the driver, a disk change must be performed
or simulated to reset the driver.

-----B-1319-----

INT 13 - FIXED DISK - PARK HEADS ON ESDI DRIVE (XT286,PS)

AH = 19h

DL = drive

Return: CF set on error

CF clear if successful

AH = status (see #00234 at AH=01h)

SeeAlso: AH=11h

-----d-1319-----

INT 13 - Future Domain SCSI CONTROLLER - REINITIALIZE DRIVE

AH = 19h

DL = hard drive ID

Return: CF set on error

AH = status code (see #00234 at AH=01h)

CF clear if successful

AH = disk type (03h = fixed disk)

CX:DX = number of 512-byte sectors

Notes: sends SCSI Read Capacity command to get number of logical blocks and

adjusts the result for 512-byte sectors
 displays either "Error in Read Capacity Command" or "nnn Bytes per sector" (nnn=256 or 512, the only sizes supported in the translation code)
 should probably be called when a removable device has its media changed
 returns the same values as AH=15h
 the TMC-950 does not support any Future Domain BIOS calls; instead, it provides a full CAM implementation (see INT 4F/AX=8100h)
 SeeAlso: AH=15h,AH=1Ah,INT 4F/AX=8100h

-----d-131A-----

INT 13 - ESDI FIXED DISK - FORMAT UNIT (PS)

AH = 1Ah
 AL = defect table entry count
 CL = format modifiers (see #00250)
 DL = drive (80h,81h)
 ES:BX -> defect table (see #00251), ignored if AL=00h

Return: CF set on error

CF clear if successful
 AH = status (see #00234 at AH=01h)

Note: if periodic interrupt selected, INT 15/AH=0Fh is called after each cylinder is formatted

SeeAlso: AH=07h,INT 15/AH=0Fh

Bitfields for ESDI format modifiers:

Bit(s) Description (Table 00250)

4 generate periodic interrupt
 3 perform surface analysis
 2 update secondary defect map
 1 ignore secondary defect map
 0 ignore primary defect map

Format of defect table entry [array]:

Offset Size Description (Table 00251)

00h 3 BYTEs relative sector address (little-endian)
 03h BYTE flags and defect count
 bit 7: last logical sector on track
 bit 6: first logical sector on track
 bit 5: last logical sector on cylinder
 bit 4: logical sectors are pushed onto next track
 bits 3-0: number of defects pushed from previous cylinder

-----d-131A-----

INT 13 - Future Domain SCSI CONTROLLER - GET SCSI PARTIAL MEDIUM CAPACITY

AH = 1Ah
CH = track (bits 8,9 in high bits of CL)
CL = sector (01h to number of sectors/track for drive)
DH = head
DL = hard drive ID

Return: CF set on error

AH = status code (see #00234 at AH=01h)
CX:DX = logical block number of last quickly-accessible block after
given block

Notes: sends SCSI Read Capacity command with the PMI bit set to obtain the
logical block address of the last block after which a substantial
delay in data transfer will be encountered (usually the last block
on the current cylinder). No translation to 512 byte sectors is
performed on the result if data is stored on the disk in other than
512 byte sectors.

the TMC-950 does not support any Future Domain BIOS calls; instead,
it provides a full CAM implementation (see INT 4F/AX=8100h)

SeeAlso: AH=15h,AH=19h"SCSI"

-----d-131B-----

INT 13 - ESDI FIXED DISK - GET MANUFACTURING HEADER

AH = 1Bh
AL = number of sectors to read
DL = drive
ES:BX -> buffer for manufacturing header (defect list)

Return: CF set on error

CF clear if successful
AH = status

Note: manufacturing header format (Defect Map Record format) can be found
in IBM 70MB, 115MB Fixed Disk Drives Technical Reference
the first sector read contains the manufacturing header with the number
of defect entries and the beginning of the defect map; the remaining
sectors contain the remainder of the defect map

-----d-131B-----

INT 13 - Future Domain SCSI CONTROLLER - GET POINTER TO SCSI DISK INFO BLOCK

AH = 1Bh
DL = hard drive ID

Return: CF set on error

AH = status code (see #00234 at AH=01h)
CF clear if successful
ES:BX -> SCSI disk information block (see #00252)

Notes: also sets a non-resettable flag which prevents some controller messages

from being displayed

the TMC-950 does not support any Future Domain BIOS calls; instead,

it provides a full CAM implementation (see INT 4F/AX=8100h)

SeeAlso: AH=18h"SCSI",AH=1Ch"SCSI"

Format of Future Domain SCSI disk information block:

Offset Size Description (Table 00252)

00h BYTE drive physical information (see #00253)
 01h WORD translated number of cylinders
 03h BYTE translated number of heads
 04h BYTE translated number of sectors per track (17, 34, or 63)
 05h BYTE drive address
 bits 0-2: logical unit number
 bits 3-5: device number
 06h BYTE 01h at initialization
 07h BYTE sense code byte 00h, or extended sense code byte 0Ch
 08h BYTE 00h
 09h BYTE 00h or extended sense code byte 02h (sense key)
 0Ah BYTE 00h
 0Bh 10 BYTES copy of Command Descriptor Block (CDB) (see #03236,#03237)
 15h DWORD translated number of sectors on device

Bitfields for Future Domain SCSI device physical information:

Bit(s) Description (Table 00253)

0 ???
 1 device uses parity
 2 256 bytes per sector instead of 512
 3 don't have capacity yet???
 4 disk is removable
 5 logical unit number is not present

-----d-131C-----

INT 13 - Future Domain SCSI CONTROLLER - GET POINTER TO FREE CONTROLLER RAM

AH = 1Ch

DL = hard drive ID (any valid SCSI hard disk)

Return: CF set on error

AH = status code (see #00234 at AH=01h)

CF clear if successful

ES:BX -> first byte of free RAM on controller

Notes: the Future Domain TMC-870 contains 1024 bytes of RAM at offsets 1800h

to 1BFFh on-board the controller for storing drive information and

controller status; ES:BX points to the first byte available for other

uses

ES contains the segment at which the controller resides; the controller's two memory-mapped I/O ports are at offsets 1C00h, 1E00h

SeeAlso: AH=1Bh"SCSI"

-----d-131C-----

INT 13 U - ESDI FIXED DISK - ???

AH = 1Ch

AL = subfunction (01h-06h)

DL = drive (80h,81h)

???

Return: ???

Note: these functions perform a controller command 0612h without DMA

SeeAlso: AX=1C08h,PORT 3510h"ESDI"

-----d-131C08-----

INT 13 U - ESDI FIXED DISK - GET COMMAND COMPLETION STATUS

AX = 1C08h

DL = drive (80h,81h)

ES:BX -> buffer for Command Complete Status Block (see #00254)

Return: CF set on error

CF clear if successful

AH = status (see #00234 at AH=01h)

SeeAlso: AX=1C09h,AX=1C0Ah,PORT 3510h"ESDI"

Format of ESDI Command Complete Status Block:

Offset Size Description (Table 00254)

00h	BYTE	07h
01h	BYTE	size of block in words (07h)
02h	BYTE	command error code (see #00255)
03h	BYTE	command status code (see #00256)
04h	BYTE	device error code, group 1 (see #00257)
05h	BYTE	device error flags, group 2 (see #00258)
06h	WORD	number of unprocessed sectors due to abnormal termination
08h	DWORD	last Relative Sector Address processed by command
0Ch	WORD	number of sectors corrected by ECC codes

(Table 00255)

Values for ESDI command error code:

00h successful

01h parameter invalid

02h unknown function

03h unsupported command
04h command cancelled
05h unknown function
06h controller diagnostics failed
07h formatting failed
08h format error in primary map
09h format error in secondary map
0Ah diagnostic failure during formatting
0Bh warning: secondary map too large during formatting
0Ch warning: non-zero defect
0Dh system checksum error during formatting
0Eh warning: incompatible device
0Fh warning: push table overflowed
10h warning: more than 15 sectors pushed to next cylinder
11h internal hardware error
12h warning: errors found while verifying sectors
13h invalid device
FFh device error

(Table 00256)

Values for ESDI command status code:

01h successful
03h successful after ECC
05h successful after retries
06h format partially completed
07h successful after ECC and retries
08h command completed with warning (see #00255)
09h abort complete
0Ah reset complete
0Bh data transfer ready (no status block)
0Ch command completed with failure (see #00257,#00258)
0Dh DMA error
0Eh command block error (see #00255)
0Fh bad attention code

SeeAlso: #00257

(Table 00257)

Values for ESDI device error code, group 1:

00h successful
01h seek fault detected by device
02h interface fault

03h sector ID not found
04h disk not formatted
05h unrecoverable ECC error
06h ECC error in sector ID
07h invalid relative sector address
08h timeout
09h sector defective
0Ah disk changed (removable media)
0Bh selection error
0Ch write protected (removable media)
0Dh write fault
0Eh read fault
0Fh no index or sector pulse
10h device not ready
11h seek error detected by adapter
12h bad format
13h volume overflow
14h data address mark not found
15h sector ID not found
16h missing device configuration data
17h first/last relative sector flags missing
18h track empty
81h timeout while waiting for stop
82h timeout while waiting for end of data transfer
84h stopped awaiting data transfer during formatting
85h timeout while waiting for head switch
86h timeout while awaiting DMA completion

SeeAlso: #00256,#00258

Bitfields for ESDI device error flags, group 2:

Bit(s) Description (Table 00258)

7-5 unused
4 ready
3 selected
2 write fault
1 on track 0
0 seek/command complete

SeeAlso: #00257

-----d-131C09-----
INT 13 U - ESDI FIXED DISK - GET DEVICE STATUS
AX = 1C09h

DL = drive (80h,81h)
 ES:BX -> buffer for Device Status Block (see #00259)
 Return: CF set on error
 CF clear if successful
 AH = status (see #00234 at AH=01h)
 SeeAlso: AX=1C08h,AX=1C0Ah,PORT 3510h"ESDI"

Format of ESDI Device Status Block:

Offset	Size	Description (Table 00259)
00h	BYTE	08h
01h	BYTE	number of words in block (09h)
02h	BYTE	error flags
03h	BYTE	unused
04h	BYTE	command error code (see #00255)
05h	BYTE	command status code (see #00256)
06h	WORD	ESDI standard status
08h	5 WORDs	ESDI vendor-specific status codes

-----d-131C0A-----

INT 13 U - ESDI FIXED DISK - GET DEVICE CONFIGURATION

AX = 1C0Ah
 DL = drive (80h,81h)
 ES:BX -> buffer for Drive Configuration Status Block (see #00260)
 Return: CF set on error
 CF clear if successful
 AH = status (see #00234 at AH=01h)

Note: device configuration format can be found in IBM ESDI Fixed Disk Drive
 Adapter/A Technical Reference

SeeAlso: AX=1C08h,AX=1C0Bh,AX=1C0Ch

Format of ESDI Drive Configuration Status Block:

Offset	Size	Description (Table 00260)
00h	BYTE	09h
01h	BYTE	number of words in block (06h)
02h	BYTE	flags
03h	BYTE	number of spare sectors per cylinder
04h	DWORD	total number of usable sectors
08h	WORD	total number of cylinders
0Ah	BYTE	tracks per cylinder
0Bh	BYTE	sectors per track

-----d-131C0B-----

INT 13 U - ESDI FIXED DISK - GET ADAPTER CONFIGURATION

```
AX = 1C0Bh
ES:BX -> buffer for Controller Configuration Status Block (see #00261)
Return: CF set on error
CF clear if successful
AH = status (see #00234 at AH=01h)
SeeAlso: AX=1C0Ch
```

Format of ESDI Controller Configuration Status Block:

```
Offset Size Description (Table 00261)
00h BYTE E9h
01h BYTE number of words in block (06h)
02h WORD unused (0000h)
04h DWORD controller microcode revision level
08h 2 WORDs unused (0000h)
```

```
-----d-131C0C-----
INT 13 U - ESDI FIXED DISK - GET POS INFORMATION
```

```
AX = 1C0Ch
ES:BX -> buffer for POS Information Status Block (see #00262)
Return: CF set on error
CF clear if successful
AH = status (see #00234 at AH=01h)
SeeAlso: AX=1C0Bh
```

Format of ESDI POS Information Status Block:

```
Offset Size Description (Table 00262)
00h BYTE EAh
01h BYTE number of words in block (05h)
02h WORD magic value FFDDh
04h BYTE POS register 3
05h BYTE POS register 2
06h BYTE POS register 5 (unused, FFh)
07h BYTE POS register 4 (unused, FFh)
08h BYTE POS register 7 (unused, FFh)
09h BYTE POS register 6 (unused, FFh)
```

```
-----d-131C0D-----
INT 13 U - ESDI FIXED DISK - ???
```

```
AX = 1C0Dh
DL = drive (80h,81h)
???
```

Return: ???

Note: invokes controller command 0614h without DMA

SeeAlso: AH=1Ch"ESDI",AX=1C0Fh,PORT 3510h"ESDI"

-----d-131C0E-----

INT 13 U - ESDI FIXED DISK - TRANSLATE RBA TO ABA

AX = 1C0Eh

CH = low 8 bits of cylinder number

CL = sector number, high two bits of cylinder number in bits 6 and 7

DH = head number

DL = drive number (80h,81h)

ES:BX -> ABA number

Return: CF set on error

CF clear if successful

AH = status (see #00234 at AH=01h)

Note: ABA (absolute block address) format can be found in IBM ESDI Adapter

Technical Reference by using its Device Configuration Status Block

SeeAlso: AX=1C08h,AX=1C0Fh,PORT 3510h"ESDI"

-----d-131C0F-----

INT 13 U - ESDI FIXED DISK - ???

AX = 1C0Fh

DL = drive (80h,81h)

???

Return: ???

Note: invokes controller command 0614h without DMA

SeeAlso: AH=1Ch"ESDI",AX=1C0Dh,AX=1C12h,PORT 3510h"ESDI"

-----d-131C12-----

INT 13 U - ESDI FIXED DISK - ???

AX = 1C12h

DL = drive (80h,81h)

???

Return: ???

Note: invokes controller command 0612h without DMA

SeeAlso: AH=1Ch"ESDI",AX=1C0Fh,PORT 3510h"ESDI"

-----c-131D-----

INT 13 - IBMCACHE.SYS - CACHE STATUS

AH = 1Dh

AL = subfunction

01h get status record

DL = drive???

Return: ES:BX -> status record (see #00263)

CF set on error

AH = error code

02h set cache status

ES:BX -> status record (see #00263)
DL = drive???
Return: CF set on error

Format of IBMCACHE.SYS status record:

Offset Size Description (Table 00263)

00h DWORD total number of read requests
04h DWORD total number of hits
08h DWORD number of physical disk reads
0Ch DWORD total number of sectors requested by physical disk reads
10h 6 BYTES ???
16h DWORD pointer to start of error list (see #00264)
1Ah DWORD pointer to end of error list
1Eh WORD ???
20h BYTE using extended memory if nonzero
21h BYTE ???
22h 4 BYTES ASCII version number
26h WORD cache size in KB
28h WORD sectors per page

Format of IBMCACHE.SYS error list:

Offset Size Description (Table 00264)

00h DWORD relative block address of bad page
04h BYTE drive
05h BYTE sector bit-map
06h WORD next error

-----d-131F-----
INT 13 - SyQuest - DOOR LATCH/DOOR BUTTON DETECT

AH = 1Fh
AL = subfunction
00h allow media removal
01h prevent media removal (lock door)
DL = drive ID (bit 7 set for hard disks)

Return: CF clear if successful

AH = 00h
CF set on error
AH = error code
00h successful
01h invalid function request
80h timeout
DDh media change requested

SeeAlso: AH=12h"SyQuest",AH=13h"SyQuest",AH=59h"SyQuest"

-----d-1320-----

INT 13 - DISK - ??? (Western Digital "Super BIOS")

AH = 20h

???

Return: ???

Notes: returns some kind of status related to whether the drive contains its
default media type

QEMM v6.00 calls INT 13/AX=2000h/DL=81h in some cases

-----b-1320-----

INT 13 U - Compaq, ATAPI Removable Media Device - GET CURRENT MEDIA FORMAT

AH = 20h

DL = drive number (00h,01h)

Return: CF clear if successful

AL = media type (see #00265)

AH = 00h

CF set on error

AH = error code

01h invalid request

30h drive does not support media sense

31h no such drive / media not present

32h non-default media / drive does not support media type

Notes: this function is supported by the 1993/3/8 Compaq ROM BIOS, but only
partially (AL is always 00h when successful) by the 1993/8/3 version

this function is also supported by some recent versions of the Phoenix
486 BIOS

this function does not seem to be supported by some Toshiba BIOSes
(at least before 1995, maybe some laptops??? with 1.44 MB floppies),
because S/DOS 1.0 contains code to bypass a call to this function,
always assuming the drive would not support media sense.

SeeAlso: AH=15h

(Table 00265)

Values for Compaq/ATAPI diskette media type:

03h 720K (1M unformatted)

04h 1.44M (2M unformatted)

06h 2.88M (4M unformatted)

0Ch 360K

0Dh 1.2M

0Eh Toshiba 3mode

0Fh NEC 3mode (1024-byte sectors)

10h ATAPI Removable Media Device

-----c-1320-----

INT 13 u - QUICKCACHE II v4.20 - DISMOUNT

AH = 20h

AL = drive (00h = A:, etc. or 7Fh for all removable drives???
or FFh for all drives)

Return: AX = status (0000h successful)

Program: QUICKCACHE II is a shareware disk cache by P.R. Glassel and
Associates, Inc.

Desc: flush any dirty buffers for the specified drive(s) and then discard
those sector buffers

SeeAlso: AH=21h"QUICKCACHE",AH=22h"QUICKCACHE",AH=28h

-----d-1321-----

INT 13 - HARD DISK - PS/1 and newer PS/2 - READ MULTIPLE DISK SECTORS

AH = 21h

AL = number of sectors to write

CH = low byte of 12-bit cylinder number

CL = starting sector (bits 0-5) and bits 8-9 of cylinder (bits 6-7)

DH = head number (bits 0-5) and bits 10-11 of cylinder (bits 6-7)

DL = drive number (80h,81h)

ES:BX -> buffer for data to be read

Return: CF clear if successful

ES:BX buffer filled

CF set on error

AH = status (see #00234 at AH=01h)

Desc: read from the disk using the Multiple Block mode available on newer
IDE drives and some hard disk controllers, which generates an
interrupt only after the end of transferring a group of sectors
rather than after each sector

Notes: must call AH=24h"PS/1" before using this function

input values in CL and DH are not range-checked

the byte at address 0040h:0074h is set to the status of the operation

SeeAlso: AH=02h, AH=22h"PS/1", AH=23h"PS/1", AH=24h"PS/1"

-----c-1321-----

INT 13 u - QUICKCACHE II v4.20 - FLUSH CACHE

AH = 21h

Return: AX = status (0000h successful)

Desc: immediately write all dirty sectors back to disk

Note: this is one out of several cache flush calls issued by the PTS-DOS 6.51
and S/DOS 1.0 kernel before rebooting. It is called with DL=80h and
conflicts with the PS/2 function of the same function number, because

the kernel does not perform Quickcache's installation check first.

SeeAlso: AH=25h"QUICKCACHE",AH=2Eh, AH=2Fh

-----d-1322-----

INT 13 - HARD DISK - PS/1 and newer PS/2 - WRITE MULTIPLE DISK SECTORS

AH = 22h

AL = number of sectors to write

CH = low byte of 12-bit cylinder number

CL = starting sector (bits 0-5) and bits 8-9 of cylinder (bits 6-7)

DH = head number (bits 0-5) and bits 10-11 of cylinder (bits 6-7)

DL = drive number (80h,81h)

ES:BX -> buffer containing data to be written

Return: CF clear if successful

CF set on error

AH = status (see #00234 at AH=01h)

Desc: write to the disk using the Multiple Block mode available on newer

IDE drives and some hard disk controllers, which generates an

interrupt only after the end of transferring a group of sectors

rather than after each sector

Notes: must call AH=24h"PS/1" before using this function

input values in CL and DH are not range-checked

the byte at address 0040h:0074h is set to the status of the operation

SeeAlso: AH=03h, AH=21h"PS/1", AH=23h"PS/1", AH=24h"PS/1"

-----c-1322-----

INT 13 u - QUICKCACHE II v4.20 - ENABLE/DISABLE CACHE

AH = 22h

AL = new state (00h disabled, 01h enabled)

Return: AX = status (0000h successful)

Note: enables/disables caching of all drives

SeeAlso: AH=2Ch, AH=2Dh, AH=32h, AH=33h, AH=A3h, AH=A4h

-----d-1323-----

INT 13 U - HARD DISK - PS/1 and newer PS/2 - SET CONTROLLER FEATURES REGISTER

AH = 23h

AL = feature number (see #00266)

DL = drive number (80h,81h)

???

Return: CF clear if successful

CF set on error

AH = status (see #00234 at AH=01h)

SeeAlso: AH=21h"PS/1", AH=22h"PS/1", AH=24h"PS/1", AH=25h"PS/1"

(Table 00266)

Values for PS/1 hard disk feature number:

01h select 8-bit data transfers instead of 16-bit
02h enable write cache
22h Write Same, user-specified area
33h disable retries
44h set number of ECC bytes for read long/write long (see AH=0Ah,AH=0Bh)
54h set cache segments
55h disable lookahead
66h disable reverting to power-on defaults
77h disable error correction
81h select 16-bit data transfers (default)
82h disable write cache
88h enable error correction (default)
99h enable retries (default)
AAh enable lookahead
BBh set ECC length for read long/write long to four bytes
CCh enable reverting to power-on defaults
DDh Write Same, entire disk

SeeAlso: #P0535

-----c-1323-----

INT 13 U - QUICKCACHE II v4.20 - GET ??? ADDRESS

AH = 23h

Return: AX = status (0000h successful)

ES = segment of ??? data

-----d-1324-----

INT 13 - HARD DISK - PS/1 and newer PS/2 - SET MULTIPLE MODE

AH = 24h

AL = number of sectors per block (2,4,8,16)

DL = drive number (80h,81h)

Return: CF clear if successful

CF set onerror

AH = status (see #00234 at AH=01h)

Desc: specify how many sectors the controller should transfer as a group
between operation-complete interrupts when using the Read Multiple
and Write Multiple functions (AH=21h,AH=22h)

Notes: set the number of sectors to 0 to disable multiple-transfer mode
the maximum value for the block size depends on the fixed disk
drive type. The value is stored in byte 15h of the fixed disk
drive parameter table that is created by POST.

the byte at address 0040h:0074h is set to status of operation.

SeeAlso: AH=21h"PS/1",AH=22h"PS/1",AH=23h"PS/1",AH=25h"PS/1"

-----c-1324-----

INT 13 u - QUICKCACHE II v4.20 - SET SECTORS

AH = 24h

BX = new number of sector buffers in cache

Return: AX = status

0000h successful

0001h failed--size adjusted

8000h cache cannot be resized while enabled

SeeAlso: AH=36h

-----d-1325-----

INT 13 - HARD DISK - PS/1 and newer PS/2 - IDENTIFY DRIVE

AH = 25h

DL = drive number (80h,81h)

ES:BX-> 512 byte buffer for reply packet

Return: CF clear if successful

CF set on error

AH = status (see #00234 at AH=01h)

buffer filled with ATA/IDE-style drive information block (see #00267)

Desc: retrieves the 256 words of drive data stored on an IDE hard disk

Notes: the byte at address 0040h:0074h is set to the status of the operation

IBM officially classifies this function as optional

SeeAlso: AH=23h"PS/1"

Format of drive information block:

Offset Size Description (Table 00267)

00h WORD general drive configuration (see #00268)

02h WORD number of cylinders

04h WORD reserved

06h WORD number of heads

08h WORD number of unformatted bytes per track

0Ah WORD number of unformatted bytes per sector

0Ch WORD number of sectors per track

0Eh 6 BYTES vendor unique

14h 20 BYTES serial number in ASCII, 0000h=not specified)

28h WORD buffer type

2Ah WORD buffer size in 512 byte increments (0000h=not specified)

2Ch WORD number of ECC bytes passed on Read/Write Long cmds

0000h = not specified

2Eh 8 BYTES firmware revision in ASCII, 0000h=not specified

36h 40 BYTES model number in ASCII, 0000h=not specified

5Eh WORD bits 15-8 Vendor Unique

```

bits 7-0 00h = Read/Write Multiple commands not implemented
      xxh = Maximum number of sectors that can be
      transferred per interrupt on Read and Write
      Multiple commands
60h WORD 0000h = cannot perform doubleword I/O
      0001h = can perform doubleword I/O
62h WORD capabilities
      bit 15-9 0=reserved
      bit 8 1=DMA Supported
      bit 7-0 Vendor Unique
64h WORD reserved
66h WORD bits 15-8 PIO data transfer cycle timing mode
      bits 7-0 Vendor Unique
68h WORD bits 15-8 DMA data transfer cycle timing mode
      bits 7-0 Vendor Unique
6Ah WORD bits 15-1 reserved
      bit 0 1=the fields reported in translation mode are valid
      0=the fields reported in translation mode may be valid
6Ch WORD number of current cylinders
6Eh WORD number of current heads
70h WORD number of current sectors per track
72h DWORD current capacity in sectors
76h WORD reserved
78h 136 BYTES not defined by ATA spec 2.6
100h 64 BYTES vendor unique
140h 96 BYTES reserved
Note: the above description is as in the ATA (AT Attachment) Specification.
SeeAlso: #P0516

```

Bitfields for general drive configuration:

```

Bit(s) Description (Table 00268)
15 0 reserved for non-magnetic drives
14 format speed tolerance gap required
13 track offset option available
12 data strobe offset option available
11 rotational speed tolerance is > 0.5%
10 disk transfer rate > 10 Mbs
9 disk transfer rate > 5Mbs but <= 10Mbs
8 disk transfer rate <= 5Mbs
7 removable cartridge drive
6 fixed drive

```



```
5 spindle motor control option implemented
4 head switch time > 15 usec
3 not MFM encoded
2 soft sectored
1 hard sectored
0 reserved (0)
-----c-1325-----
INT 13 u - QUICKCACHE II v4.20 - SET FLUSH INTERVAL
    AH = 25h
    BX = interval
Return: AX = status (0000h successful)
Desc: specify how often the cache should write dirty buffers to disk when
    buffered writes are enabled
SeeAlso: AH=21h"QUICKCACHE",AH=2Ch,AH=2Eh
-----c-1326-----
INT 13 U - QUICKCACHE II v4.20 - UNINSTALL
    AH = 26h
Return: AX = status
    0000h successful
    0001h-00FFh interrupt vector which was hooked by another TSR
SeeAlso: AH=27h
-----c-1327--BX0000-----
INT 13 u - QUICKCACHE II v4.20 - INSTALLATION CHECK
    AH = 27h
    BX = 0000h
Return: AX = 0000h if installed
    BX nonzero if installed
    BH = major version
    BL = binary minor version
Program: QUICKCACHE II is a shareware disk cache by P.R. Glassel and
    Associates, Inc.
SeeAlso: AH=26h,AH=A0h,INT 16/AX=FFA5h/CX=1111h
-----c-1328-----
INT 13 U - QUICKCACHE II v4.20 - SET AUTOMATIC DISMOUNT
    AH = 28h
    AL = new state (00h disabled, 01h enabled)
Return: AX = status (0000h successful)
SeeAlso: AH=20h"QUICKCACHE"
-----c-1329-----
INT 13 U - QUICKCACHE II v4.20 - NOP
    AH = 29h
```

Return: AX = 0000h

-----c-132A-----

INT 13 u - QUICKCACHE II v4.20 - SET BUFFER SIZE

AH = 2Ah

AL = buffer size (1-30)

Return: AX = status (0000h successful)

Desc: specify the number of cache sector buffers to dedicate to buffered read
and write operations

SeeAlso: AH=2Ch,AH=2Dh,AH=39h,AH=3Ah

-----c-132B-----

INT 13 U - QUICKCACHE II v4.20 - DRIVE ACCESS SOUNDS

AH = 2Bh

AL = new state (00h disabled, 01h enabled)

Return: AX = status (0000h successful)

-----c-132C-----

INT 13 u - QUICKCACHE II v4.20 - SET BUFFERED WRITES

AH = 2Ch

AL = new state (00h disabled, 01h enabled)

Return: AX = status (0000h successful)

Desc: specify whether the cache should delay disk writes

Note: this function enables or disables delayed writes for all drives; use

AH=38h to change a single drive

SeeAlso: AH=25h"QUICKCACHE",AH=2Dh,AH=2Eh,AH=38h

-----c-132D-----

INT 13 u - QUICKCACHE II v4.20 - SET BUFFERED READ

AH = 2Dh

AL = new state (00h disabled, 01h enabled)

Return: AX = status (0000h successful)

Desc: specify whether the cache should attempt to read ahead of actual
requests

Note: this function enables or disables read-ahead for all drives; use AH=37h
to change a single drive

SeeAlso: AH=2Ch,AH=37h

-----c-132E-----

INT 13 u - QUICKCACHE II v4.20 - SET FLUSH COUNT

AH = 2Eh

BX = flush count

Return: AX = status (0000h successful)

Desc: specify how many dirty sectors the cache should write after each flush
interval (see AH=25h"QUICKCACHE") when buffered writes are enabled

SeeAlso: AH=21h"QUICKCACHE",AH=25h"QUICKCACHE",AH=2Ch

-----c-132F-----

INT 13 - QUICKCACHE II v4.20 - FORCE IMMEDIATE INCREMENTAL FLUSH

AH = 2Fh

Return: AX = status (0000h successful)

Desc: immediately flush up to "flushcount" dirty sectors to disk as if the
flush interval had expired

SeeAlso: AH=21h"QUICKCACHE"

-----c-1330-----

INT 13 u - QUICKCACHE II v4.20 - GET INFO

AH = 30h

AL = what to get

00h system info (see #00269)

01h drive info (see #00270)

02h access frequency (array of 30 words)

03h drive index

(array of 32 bytes indicating BIOS drive for DOS drive)

DS:DX -> buffer for info

Return: AX = status (0000h successful, 8000h invalid info specifier)

Program: QUICKCACHE II is a shareware disk cache by P.R. Glassel and
Associates, Inc.

Format of QUICKCACHE II system info:

Offset Size Description (Table 00269)

00h	BYTE	flag: cache enabled
01h	BYTE	flag: buffered writes enabled
02h	BYTE	flag: buffered reads enabled
03h	BYTE	flag: sounds enabled
04h	BYTE	flag: autodismount enabled
05h	BYTE	???
06h	BYTE	flag: ???
07h	BYTE	flag: ???
08h	BYTE	flag: "em_assigned"
09h	BYTE	flag: emulated EMS
0Ah	BYTE	single sector bonus
0Bh	BYTE	"sticky_max"
0Ch	BYTE	write sector bonus
0Dh	BYTE	bonus threshold
0Eh	WORD	flush interval
10h	WORD	flush count
12h	WORD	reserve pool size
14h	WORD	remaining space in reserve pool

16h WORD required free memory
 18h WORD total cache sectors
 1Ah WORD dirty cache sectors
 1Ch BYTE trace buffer size
 1Dh BYTE reserved (padding)

SeeAlso: #00270

Format of QUICKCACHE II drive info [16-element array, one element]:

Offset Size Description (Table 00270)

00h BYTE DOS drive number
 01h BYTE BIOS drive number
 02h BYTE maximum sector number
 03h BYTE maximum head number
 04h BYTE read buffer size
 05h BYTE write buffer size
 06h BYTE last status
 07h BYTE flag: enabled
 08h BYTE flag: buffered write enabled
 09h BYTE flag: buffered read enabled
 0Ah BYTE flag: in use (drive info is valid)
 0Bh BYTE flag: cylinder flush
 0Ch BYTE reserved (padding)
 0Dh BYTE sectors per track
 0Eh WORD sector size
 10h WORD sectors assigned
 12h WORD dirty sectors
 14h WORD reserved sectors
 16h WORD number of read errors
 18h WORD number of write errors
 1Ah DWORD "rio_count"
 1Eh DWORD number of cache misses
 22h DWORD "wio_count"
 26h DWORD "dio_count"

SeeAlso: #00269

-----c-1331-----
 INT 13 U - QUICKCACHE II v4.20 - RESERVE MEMORY
 AH = 31h
 BX = number of paragraphs of conventional memory to reserve for apps

Return: AX = status (0000h successful)

-----c-1332-----
 INT 13 U - QUICKCACHE II v4.20 - ENABLE CACHING FOR SPECIFIC DRIVE

```
AH = 32h
AL = drive number (00h=A:)
Return: AX = status (0000h successful)
SeeAlso: AH=22h"QUICKCACHE",AH=33h
-----c-1333-----
INT 13 U - QUICKCACHE II v4.20 - DISABLE CACHING FOR SPECIFIC DRIVE
  AH = 33h
  AL = drive number (00h=A:)
Return: AX = status (0000h successful)
SeeAlso: AH=22h"QUICKCACHE",AH=32h
-----c-1334-----
INT 13 U - QUICKCACHE II v4.20 - SECTOR LOCKING
  AH = 34h
  AL = function
    00h end sector locking/unlocking
    01h lock all accessed sectors into cache
    02h unlock all accessed sectors and discard from cache
Return: AX = status (0000h successful)
SeeAlso: AH=20h"QUICKCACHE",AH=35h
-----c-1335-----
INT 13 U - QUICKCACHE II v4.20 - SET LOCK POOL SIZE
  AH = 35h
  BX = number of sectors in lock pool
Return: AX = status (0000h successful)
Desc: specify the number of cache sector buffers which may be dedicated to
      data locked into the cache
SeeAlso: AH=34h
-----c-1336-----
INT 13 U - QUICKCACHE II v4.20 - SET TRACE BUFFER SIZE
  AH = 36h
  AL = new size of trace buffer
Return: AX = status (0000h successful)
Note: called with AL=05h during an INT 13/AH=24h"QUICKCACHE" call
SeeAlso: AH=24h"QUICKCACHE"
-----c-1337-----
INT 13 U - QUICKCACHE II v4.20 - SET BUFFERED READS FOR SPECIFIC DRIVE
  AH = 37h
  AL = new state (00h disabled, else enabled)
  DL = drive number (00h = A:)
Return: AX = status (0000h successful)
SeeAlso: AH=2Dh,AH=38h
```

-----c-1338-----

INT 13 U - QUICKCACHE II v4.20 - SET BUFFERED WRITES FOR SPECIFIC DRIVE

AH = 38h

AL = new state (00h disabled, else enabled)

DL = drive number (00h = A:)

Return: AX = status (0000h successful)

SeeAlso: AH=2Ch,AH=37h

-----c-1339-----

INT 13 U - QUICKCACHE II v4.20 - SET READ BUFFER SIZE FOR SPECIFIC DRIVE

AH = 39h

AL = new size of read buffer

DL = drive number (00h = A:)

Return: AX = status (0000h successful)

Program: QUICKCACHE II is a shareware disk cache by P.R. Glassel and

Associates, Inc.

SeeAlso: AH=2Ah,AH=3Ah

-----c-133A-----

INT 13 U - QUICKCACHE II v4.20 - SET WRITE BUFFER SIZE FOR SPECIFIC DRIVE

AH = 3Ah

AL = new size of write buffer

DL = drive number (00h = A:)

Return: AX = status (0000h successful)

SeeAlso: AH=2Ah,AH=39h

-----c-133B-----

INT 13 U - QUICKCACHE II v4.20 - ENABLE/DISABLE ???

AH = 3Bh

AL = new state of ??? (01h enabled, else disabled)

Return: AX = status (0000h successful)

Note: is affected by the flag reported at offset 05h of the system info

returned by AH=30h, and sets the flag at offset 06h

SeeAlso: AH=3Ch

-----c-133C-----

INT 13 U - QUICKCACHE II v4.20 - ENABLE/DISABLE ???

AH = 3Ch

AL = new state of ??? (01h enabled, else disabled)

Return: AX = status (0000h successful)

Note: is affected by the flag reported at offset 05h of the system info

returned by AH=30h, and sets the flag at offset 07h

SeeAlso: AH=3Bh

-----c-133D-----

INT 13 U - QUICKCACHE II v4.20 - ENABLE/DISABLE CYLINDER FLUSH FOR DRIVE

```
AH = 3Dh
AL = new state (01h enabled, else disabled)
DL = drive number (00h = A:)
Return: AX = status (0000h successful)
-----c-133E-----
INT 13 U - QUICKCACHE II v4.20 - SET SINGLE-SECTOR BONUS
  AH = 3Eh
  AL = new value for bonus
Return: AX = status (0000h successful)
Desc: specify the bonus score to give to single-sector transfers in order to
      keep those sectors in the cache longer
-----c-133F-----
INT 13 U - QUICKCACHE II v4.20 - SET BONUS THRESHOLD
  AH = 3Fh
  AL = new value for bonus threshold
Return: AX = status (0000h successful)
-----c-1340-----
INT 13 U - QUICKCACHE II v4.20 - SET "sticky_max"
  AH = 40h
  AL = new value for "sticky_max"
Return: AX = status (0000h successful)
SeeAlso: AH=41h"QUICKCACHE"
-----d-1341--BX55AA-----
INT 13 - IBM/MS INT 13 Extensions - INSTALLATION CHECK
  AH = 41h
  BX = 55AAh
  DL = drive (80h-FFh)
Return: CF set on error (extensions not supported)
  AH = 01h (invalid function)
  CF clear if successful
  BX = AA55h if installed
  AH = major version of extensions
  01h = 1.x
  20h = 2.0 / EDD-1.0
  21h = 2.1 / EDD-1.1
  30h = EDD-3.0
  AL = internal use
  CX = API subset support bitmap (see #00271)
  DH = extension version (v2.0+ ??? -- not present in 1.x)
Note: the Phoenix Enhanced Disk Drive Specification v1.0 uses version 2.0 of
the INT 13 Extensions API
```

SeeAlso: AH=42h"INT 13 Ext",AH=48h"INT 13 Ext"

Bitfields for IBM/MS INT 13 Extensions API support bitmap:

Bit(s) Description (Table 00271)

- 0 extended disk access functions (AH=42h-44h,47h,48h) supported
- 1 removable drive controller functions (AH=45h,46h,48h,49h,INT 15/AH=52h) supported
- 2 enhanced disk drive (EDD) functions (AH=48h,AH=4Eh) supported
extended drive parameter table is valid (see #00273,#00278)
- 3-15 reserved (0)

-----c-1341-----

INT 13 U - QUICKCACHE II v4.20 - SAVE/RESTORE ???

AH = 41h

AL = direction

01h save to file

else restore from file

ES:DI -> 1024-byte buffer for ???

Return: AX = status (0000h successful, 8000h failed)

Program: QUICKCACHE II is a shareware disk cache by P.R. Glassel and Associates, Inc.

SeeAlso: AH=40h"QUICKCACHE"

-----d-1342-----

INT 13 - IBM/MS INT 13 Extensions - EXTENDED READ

AH = 42h

DL = drive number

DS:SI -> disk address packet (see #00272)

Return: CF clear if successful

AH = 00h

CF set on error

AH = error code (see #00234)

disk address packet's block count field set to number of blocks successfully transferred

SeeAlso: AH=02h,AH=41h"INT 13 Ext",AH=43h"INT 13 Ext"

Format of disk address packet:

Offset Size Description (Table 00272)

- 00h BYTE size of packet (10h or 18h)
- 01h BYTE reserved (0)
- 02h WORD number of blocks to transfer (max 007Fh for Phoenix EDD)
- 04h DWORD -> transfer buffer
- 08h QWORD starting absolute block number


```
(for non-LBA devices, compute as
(Cylinder*NumHeads + SelectedHead) * SectorPerTrack +
SelectedSector - 1
10h QWORD (EDD-3.0, optional) 64-bit flat address of transfer buffer;
used if DWORD at 04h is FFFFh:FFFFh
```

```
-----N-134257DX1234-----
```

```
INT 13 U - Beame&Whiteside BWLPD - INSTALLATION CHECK
```

```
AX = 4257h ("BW")
```

```
DX = 1234h
```

```
Return: BX = 414Ch if installed
```

```
Program: BWLPD is the printer daemon from the BW-NFS package
```

```
SeeAlso: INT 62/AH=00h"ETHDEV"
```

```
-----d-1343-----
```

```
INT 13 - IBM/MS INT 13 Extensions - EXTENDED WRITE
```

```
AH = 43h
```

```
AL = write flags
```

```
---v1.0,2.0---
```

```
bit 0: verify write
```

```
bits 7-1 reserved (0)
```

```
---v2.1+ ---
```

```
00h,01h write without verify
```

```
02h write with verify
```

```
DL = drive number
```

```
DS:SI -> disk address packet (see #00272)
```

```
Return: CF clear if successful
```

```
AH = 00h
```

```
CF set on error
```

```
AH = error code (see #00234)
```

```
disk address packet's block count field set to number of blocks
```

```
successfully transferred
```

```
Note: the BIOS returns CF set/AH=01h (invalid function) if verify is
requested but not supported
```

```
SeeAlso: AH=03h,AH=41h"INT 13 Ext",AH=42h"INT 13 Ext",AH=44h
```

```
-----d-1344-----
```

```
INT 13 - IBM/MS INT 13 Extensions - VERIFY SECTORS
```

```
AH = 44h
```

```
DL = drive number
```

```
DS:SI -> disk address packet (see #00272)
```

```
Return: CF clear if successful
```

```
AH = 00h
```

```
CF set on error
```

AH = error code (see #00234)
disk address packet's block count field set to number of blocks
successfully verified

SeeAlso: AH=04h,AH=41h"INT 13 Ext",AH=42h"INT 13 Ext",AH=47h

-----d-1345-----

INT 13 - IBM/MS INT 13 Extensions - LOCK/UNLOCK DRIVE

AH = 45h
AL = operation
00h lock media in drive
01h unlock media
02h check lock status

DL = drive number

Return: CF clear if successful

AH = 00h
AL = lock state (00h = unlocked)

CF set on error

AH = error code (see #00234)

Notes: this function is required to be supported for any removable drives
numbered 80h or higher

a device may be locked even if no media is present in the drive
up to 255 locks may be placed on a drive, and the media will not
be physically unlocked until all locks have been removed

SeeAlso: AH=41h"INT 13 Ext",AH=46h,AH=49h,INT 15/AH=52h"INT 13 Extensions"

-----d-1346-----

INT 13 - IBM/MS INT 13 Extensions - EJECT MEDIA

AH = 46h
AL = 00h (reserved)
DL = drive number

Return: CF clear if successful

AH = 00h
CF set on error
AH = error code (see #00234)

SeeAlso: AH=49h,INT 15/AH=52h"INT 13 Extensions"

-----d-1347-----

INT 13 - IBM/MS INT 13 Extensions - EXTENDED SEEK

AH = 47h
DL = drive number
DS:SI -> disk address packet (see #00272)

Return: CF clear if successful

AH = 00h
CF set on error

AH = error code (see #00234)

Note: this function initiates the seek, and may return before the seek actually completes

SeeAlso: AH=0Ch,AH=42h"INT 13 Ext"

-----d-1348-----

INT 13 - IBM/MS INT 13 Extensions - GET DRIVE PARAMETERS

AH = 48h

DL = drive (80h-FFh)

DS:SI -> buffer for drive parameters (see #00273)

Return: CF clear if successful

AH = 00h

DS:SI buffer filled

CF set on error

AH = error code (see #00234)

BUGS: several different Compaq BIOSes incorrectly report high-numbered drives (such as 90h, B0h, D0h, and F0h) as present, giving them the same geometry as drive 80h; as a workaround, scan through disk numbers, stopping as soon as the number of valid drives encountered equals the value in 0040h:0075h

Dell machines using PhoenixBIOS 4.0 Release 6.0 fail to correctly handle this function if the flag word at DS:[SI+2] is not 0000h on entry

SeeAlso: AH=08h,AH=41h,AH=49h,MEM 0040h:0075h

Format of IBM/MS INT 13 Extensions drive parameters:

Offset Size Description (Table 00273)

00h WORD (call) size of buffer

(001Ah for v1.x, 001Eh for v2.x, 42h for v3.0)

(ret) size of returned data

02h WORD information flags (see #00274)

04h DWORD number of physical cylinders on drive

08h DWORD number of physical heads on drive

0Ch DWORD number of physical sectors per track

10h QWORD total number of sectors on drive

18h WORD bytes per sector

---v2.0+ ---

1Ah DWORD -> EDD configuration parameters (see #00278)

FFFFh:FFFFh if not available

---v3.0 ---

1Eh WORD signature BEDDh to indicate presence of Device Path info

20h BYTE length of Device Path information, including signature and this

```

    byte (24h for v3.0)
21h  3 BYTES reserved (0)
24h  4 BYTES ASCIZ name of host bus ("ISA" or "PCI")
28h  8 BYTES ASCIZ name of interface type
    "ATA"
    "ATAPI"
    "SCSI"
    "USB"
    "1394" IEEE 1394 (FireWire)
    "FIBRE" Fibre Channel
30h  8 BYTES Interface Path (see #00275)
38h  8 BYTES Device Path (see #00276)
40h  BYTE reserved (0)
41h  BYTE checksum of bytes 1Eh-40h (two's complement of sum, which makes
    the 8-bit sum of bytes 1Eh-41h equal 00h)

```

Note: if the size is less than 30 on call, the final DWORD will not be returned by a v2.x implementation; similarly for the Device Path info

SeeAlso: #00277,#03196

Bitfields for IBM/MS INT 13 Extensions information flags:

Bit(s) Description (Table 00274)

```

0  DMA boundary errors handled transparently
1  cylinder/head/sectors-per-track information is valid
2  removable drive
3  write with verify supported
4  drive has change-line support (required if drive >= 80h is removable)
5  drive can be locked (required if drive >= 80h is removable)
6  CHS information set to maximum supported values, not current media
15-7 reserved (0)

```

Note: bits 4-6 are only valid if bit 2 is set

SeeAlso: #00273

Format of EDD v3.0 Interface Path:

Offset Size Description (Table 00275)

---ISA---

```

00h  WORD  16-bit base address
02h  6 BYTES reserved (0)

```

---PCI---

```

00h  BYTE  PCI bus number
01h  BYTE  PCI device number
02h  BYTE  PCI function number

```

03h 5 BYTES reserved (0)

SeeAlso: #00273,#00276

Format of EDD v3.0 Device Path:

Offset Size Description (Table 00276)

---ATA---

00h BYTE flag: 00h = master, 01h = slave

01h 7 BYTES reserved (0)

---ATAPI---

00h BYTE flag: 00h = master, 01h = slave

01h BYTE logical unit number

02h 6 BYTES reserved (0)

---SCSI---

00h BYTE logical unit number

01h 7 BYTES reserved (0)

---USB---

00h BYTE to be determined

01h 7 BYTES reserved (0)

---IEEE1394---

00h QWORD 64-bit FireWire General Unique Identifier (GUID)

---FibreChannel---

00h QWORD Word Wide Number (WWN)

SeeAlso: #00273,#00275

Format of Phoenix Enhanced Disk Drive Spec translated drive parameter table:

Offset Size Description (Table 00277)

00h WORD number of cylinders

02h BYTE number of heads

03h BYTE A0h (signature indicating translated table)

04h BYTE number of physical sectors per track

05h WORD starting write precompensation cylinder number

07h BYTE reserved

08h BYTE control byte (see #03198 at INT 41"DISK 0")

09h WORD number of physical cylinders

0Bh BYTE number of physical heads

0Ch WORD cylinder number of landing zone

0Eh BYTE number of logical sectors per track

0Fh BYTE checksum

Program: the Phoenix Enhanced Disk Drive Specification is an addition to the
IBM/MS INT 13 extensions

SeeAlso: #00278,#03196

Format of Phoenix Enhanced Disk Drive Spec Fixed Disk Parameter Table:

Offset	Size	Description (Table 00278)
00h	WORD	physical I/O port base address
02h	WORD	disk-drive control port address
04h	BYTE	drive flags (see #00279)
05h	BYTE	proprietary information bits 7-4 reserved (0) bits 3-0: Phoenix proprietary (used by BIOS)
06h	BYTE	IRQ for drive (bits 3-0; bits 7-4 reserved and must be 0)
07h	BYTE	sector count for multi-sector transfers
08h	BYTE	DMA control bits 7-4: DMA type (0-2) as per ATA-2 specification bits 3-0: DMA channel
09h	BYTE	programmed I/O control bits 7-4: reserved (0) bits 3-0: PIO type (1-4) as per ATA-2 specification
0Ah	WORD	drive options (see #00280)
0Ch	2 BYTES	reserved (0)
0Eh	BYTE	extension revision level (high nybble=major, low nybble=minor) (currently 10h for v1.0 and 11h for v1.1-3.0)
0Fh	BYTE	2's complement checksum of bytes 00h-0Eh 8-bit sum of all bytes 00h-0Fh should equal 00h

Note: this structure is also called the Device Parameter Table Extension (DPTE)

SeeAlso: #00277

Bitfields for Phoenix Enhanced Disk Drive Spec drive flags:

Bit(s)	Description (Table 00279)
7	reserved (1)
6	LBA enabled
5	reserved (1)
4	drive is slave
3-0	reserved (0)

SeeAlso: #00278,#00280

Bitfields for Phoenix Enhanced Disk Drive Spec drive options:

Bit(s)	Description (Table 00280)
0	fast PIO enabled
1	fast DMA access enabled
2	block PIO (multi-sector transfers) enabled

3 CHS translation enabled
 4 LBA translation enabled
 5 removable media
 6 ATAPI device (CD-ROM)
 7 32-bit transfer mode

---v1.1+ ---

8 ATAPI device uses DRQ to signal readiness for packet command
 (must be 0 if bit 6 is 0)
 10-9 translation type (must be 00 if bit 3 is 0)
 00 Phoenix bit-shifting translation
 01 LBA-assisted translation
 10 reserved
 11 proprietary translation

---v3.0---

11 Ultra DMA access enabled
 15-12 reserved (0)

SeeAlso: #00278,#00279

-----d-1349-----

INT 13 - IBM/MS INT 13 Extensions - EXTENDED MEDIA CHANGE

AH = 49h
 DL = drive number

Return: CF clear if media has not changed

AH = 00h

CF set if media may have changed

AH = 06h (see #00234)

Note: unlike AH=16h, any drive number may be specified

SeeAlso: AH=16h,AH=41h"INT 13 Ext",AH=46h

-----d-134A-----

INT 13 - Bootable CD-ROM - INITIATE DISK EMULATION

AH = 4Ah
 AL = 00h
 DS:SI -> specification packet (see #00281)

Return: CF clear if successful

CF set on error (drive will not be in emulation mode)

AX = return codes

SeeAlso: AH=48h,AX=4B00h,AH=4Ch,AH=4Dh

Format of Bootable CD-ROM Specification Packet:

Offset	Size	Description (Table 00281)
00h	BYTE	size of packet in bytes (13h)
01h	BYTE	boot media type (see #00282)

```

02h BYTE drive number
    00h floppy image
    80h bootable hard disk
    81h-FFh nonbootable or no emulation
03h BYTE CD-ROM controller number
04h DWORD Logical Block Address of disk image to emulate
08h WORD device specification (see also #00282)
    (IDE) bit 0: drive is slave instead of master
    (SCSI) bits 7-0: LUN and PUN
        bits 15-8: bus number
0Ah WORD segment of 3K buffer for caching CD-ROM reads
0Ch WORD load segment for initial boot image
    if 0000h, load at segment 07C0h
0Eh WORD number of 512-byte virtual sectors to load
    (only valid for AH=4Ch)
10h BYTE low byte of cylinder count (for INT 13/AH=08h)
11h BYTE sector count, high bits of cylinder count (for INT 13/AH=08h)
12h BYTE head count (for INT 13/AH=08h)
SeeAlso: #00283,AH=08h

```

Bitfields for Bootable CD-ROM boot media type:

```

Bit(s) Description (Table 00282)
3-0 media type
    0000 no emulation
    0001 1.2M diskette
    0010 1.44M diskette
    0011 2.88M diskette
    0100 hard disk (drive C:)
    other reserved
5-4 reserved (0)
6 image contains ATAPI driver
7 image contains SCSI driver(s)

```

SeeAlso: #00281

-----d-134B00-----

INT 13 - Bootable CD-ROM - TERMINATE DISK EMULATION

```

AX = 4B00h
DL = drive number or 7Fh to terminate all emulations
DS:SI -> empty specification packet (see #00281)
Return: CF clear if successful
    CF set on error (drive will still be in emulation mode)
AX = return codes

```


DS:SI specification packet filled
SeeAlso: AH=48h,AH=4Ah,AX=4B00h,AH=4Ch,AH=4Dh
-----d-134B01-----
INT 13 - Bootable CD-ROM - GET STATUS
AX = 4B01h
DL = drive number
DS:SI -> empty specification packet (see #00281)
Return: CF clear if successful
CF set on error
AX = return codes
DS:SI specification packet filled
Note: same as AX=4B00h, but does not terminate emulation
SeeAlso: AH=48h,AH=4Ah,AX=4B00h,AH=4Ch,AH=4Dh
-----d-134C-----
INT 13 - Bootable CD-ROM - INITIATE DISK EMULATION AND BOOT
AH = 4Ch
AL = 00h
DS:SI -> specification packet (see #00281)
Return: never, if successful
CF set (error while attempting to boot)
AX = error codes
SeeAlso: AH=48h,AH=4Ah,AX=4B00h,AH=4Dh
-----d-134D00-----
INT 13 - Bootable CD-ROM - RETURN BOOT CATALOG
AX = 4D00h
DS:SI -> command packet (see #00283)
Return: CF clear if successful
CF set on error
AX = return codes
SeeAlso: AH=48h,AH=4Ah,AX=4B00h,AH=4Ch

Format of Bootable CD-ROM "get boot catalog" command packet:
Offset Size Description (Table 00283)
00h BYTE size of packet in bytes (08h)
01h BYTE number of sectors of boot catalog to read
02h DWORD -> buffer for boot catalog
06h WORD first sector in boot catalog to transfer
SeeAlso: #00281
-----d-134E-----
INT 13 - IBM/MS INT 13 Extensions v2.1+ - SET HARDWARE CONFIGURATION
AH = 4Eh

AL = function
 00h enable prefetch
 01h disable prefetch
 02h set maximum PIO transfer mode
 03h set PIO mode 0
 04h set default PIO transfer mode
 05h enable INT 13h DMA maximum mode
 06h disable INT 13h DMA

DL = drive number

Return: CF clear if successful

AH = 00h

AL = status

00h command was safe (only affected specified drive)

01h other devices are affected

CF set on error

AH = error code (see #00234)

Note: DMA and PIO modes are mutually exclusive, so selecting DMA disables

PIO (for either the specified device or all devices on that controller), and selecting PIO disables DMA

SeeAlso: AH=41h"INT 13 Extensions",AX=5001h"Enhanced Disk Drive"

-----d-135001-----

INT 13 - Enhanced Disk Drive Spec v3.0 - SEND PACKET COMMAND

AX = 5001h

DL = drive number

ES:BX -> command packet (see #00284)

Return: CF clear if successful

AH = 00h

CF set on error

AH = error code

Desc: send data to and from a serial packet-oriented device, such as IEEE1394 and USB

SeeAlso: AH=41h"INT 13 Extensions",AH=4Eh

Format of Enhanced Disk Drive Spec v3.0 command packet:

Offset Size Description (Table 00284)

00h WORD signature B055h

02h BYTE length of packet in bytes

03h BYTE reserved (0)

04h N BYTES formatted packet data

-----v-135001-----

INT 13 - VIRUS - "Andropinis" - INSTALLATION CHECK

```
AX = 5001h
Return: AX = 0150h if resident
SeeAlso: AX=FD50h"VIRUS",INT 21/AX=0B56h
-----v-135342CX0001-----
INT 13 - ScanBoot - INSTALLATION CHECK
  AX = 5342h ("SB")
  CX = 0001h
  DX = 0000h
Return: CF clear if ScanBoot installed
  AX = 0000h
  CX = serial number ("SW" if shareware release)
  DX = version
  BX,SI,ES destroyed
Program: ScanBoot is a virus-detection TSR by PanSoft
-----d-135501-----
INT 13 - Seagate ST01/ST02 - Inquiry
  AX = 5501h
  DH = number of bytes to transfer
  DL = drive ID (80h, 81h, ...)
  ES:BX -> buffer for results
Return: ES:BX buffer filled with the Inquiry results
Notes: the ST01/ST02 BIOS does not return any success/failure indication,
  so all commands must be assumed to have been successful
  the ST01/ST02 BIOS always maps its drives after the previous BIOS
  drives without changing the BIOS drive count at 0040h:0075h
  this command is identical to the SCSI Inquiry command
-----d-135502-----
INT 13 - Seagate ST01/ST02 - RESERVED
  AX = 5502h
-----d-135503-----
INT 13 - Seagate ST01/ST01 - Set Device Type Qualifier (DTQ)
  AX = 5503h
  DH = DTQ byte (see #00285)
  DL = drive ID (80h, 81h, ...)
Return: nothing
```

Bitfields for DTQ byte:

Bit(s) Description (Table 00285)

```
7 reserved
6 SCSI drive attached
5 reserved
```

```
4 selected drive is ST225N/NP (Paired)
3 selected drive is ST225N
2 Host Adapter checks parity on the selected drive
1 selected drive has been installed
0 Seagate installation software present
-----d-135504-----
INT 13 U - Seagate - ??? - RETURN IDENTIFICATION
  AX = 5504h
  DX = drive (bit 7 set for hard disk)
Return: CF clear if successful
  AX = 4321h if ST01/ST02h
  AX = 4322h if ??? Seagate controller
  CF set on error
SeeAlso: AX=5505h,AX=5514h
-----d-135504-----
INT 13 - Seagate ST01/ST02 - RETURN IDENTIFICATION
  AX = 5504h
  DL = drive ID (80h, 81h, ...)
Return: AX = 4321h
  BL = selected drive number (00h, 01h)
  BH = number of drives attached to Host Adapter (max. 2)
-----d-135505-----
INT 13 - Seagate - ??? - PARK HEADS
  AX = 5505h
  DX = drive (bit 7 set for hard disk)
Return: CF clear if successful
  CF set on error
SeeAlso: AX=5504h,AX=5515h
-----d-135505-----
INT 13 - Seagate ST01/ST02 - PARK HEADS
  AX = 5505h
  DL = drive ID (80h, 81h, ...)
  DH = subfunction
    00h park heads (SCSI Stop command)
    01h un-park heads (SCSI Start command)
Return: nothing
-----d-135506-----
INT 13 - Seagate ST01/ST02 - SCSI Bus Parity
  AX = 5506h
  DL = drive ID (80h, 81h, ...)
  DH = subfunction
```

00h disable parity check
01h enable parity check
02h return current parity setting

Return: AL = status

00h parity checking disabled
01h parity checking enabled

-----d-135507-----

INT 13 - Seagate ST01/ST02 - RESERVED FUNCTIONS

AX = 5507h to 550Dh

Note: officially listed as "reserved"

-----d-135514-----

INT 13 U - Seagate - ???

AX = 5514h

DX = drive (bit 7 set for hard disk)

Return: CF clear if successful

CF set on error

AX = return value (FEBEh, FEBFh, FEDAh, FEDBh)

SeeAlso: AX=5504h,AX=5515h

-----d-135515-----

INT 13 U - Seagate - PARK HEADS???

AX = 5515h

DX = drive (bit 7 set for hard disk)

Return: CF clear if successful

CF set on error

Note: appears to be identical to AX=5505h

SeeAlso: AX=5504h,AX=5505h

-----d-1359-----

INT 13 - SyQuest - Generic SCSI pass through

AH = 59h

CX = HOST_ID, 0-based

DX = 80h

ES:BX pointer to SCSI structure (see #00286)

Return: CF clear

AH = 95h

SeeAlso: AH=12h"SyQuest",AH=13h"SyQuest",AH=1Fh"SyQuest"

Format of SyQuest SCSI structure:

Offset Size Description (Table 00286)

00h WORD opcode (see #00287)

02h BYTE target's SCSI ID

03h BYTE target's logical unit number

04h BYTE data direction (00h no data xfer, 01h data in, FFh data out)

05h BYTE host status

00h successful

01h selection time out

02h data over-run or under-run

06h BYTE target status at command completion

00h successful

02h check status

08h busy

07h BYTE command data block length

08h DWORD request data length

0Ch DWORD result data length (actual length of data transferred)

10h DWORD -> CDB (see #03236,#03237,#03238)

14h DWORD -> data buffer

Note: The handler does not perform a 'Request Sense' command if there was an error

(Table 00287)

Values for SCSI opcode:

00h verify interface

clears carry flag and returns if function is available

01h returns the ID of the INT 13h Handler in a NULL terminated string of length less than 40 byte including the terminator.

The string is stored in the buffer pointed by p_buf.

02h device mapping info. The caller provides a one byte buffer.

The handler stores the Int 13h Device ID (80h or above) in the buffer.

It stores 0 if that target does not exists.

03h execute SCSI command

04h device reset

05h SCSI bus reset

SeeAlso: #00286

-----d-1370-----

INT 13 - Priam EDVR.SYS DISK PARTITIONING SOFTWARE???

AH = 70h

???

Return: ???

Note: Priam's EDISK.EXE (FDISK replacement) and EFMT.EXE (low-level formatting program) make this call, presumably to EDVR.SYS (the partitioning driver)

SeeAlso: AH=ADh

-----1375-----

INT 13 - ???

AH = 75h

???

Return: AH = ???

???

Note: intercepted by PC-Cache (v5.1 only)

-----1376-----

INT 13 - ???

AH = 76h

???

Return: AH = ???

???

Note: intercepted by PC-Cache (v5.1 only)

-----c-137B00-----

INT 13 - NOW! v3.05 - GET INFORMATION

AX = 7B00h

CX:DX -> 1F8h-byte buffer for information record (see #00288)

Return: AX = 0000h

BX = segment of main resident code

ES = ???

Program: NOW! is a disk cache by Vertissoft Systems, Inc.

SeeAlso: AX=7B02h,AH=EFh

Format of NOW! information record:

Offset Size Description (Table 00288)

00h 80 BYTES name of directory from which NOW! was started

50h 424 BYTES ???

81h ? BYTES array of bytes for ???

F7h 250 BYTES array of 25 entries, one per drive???

Offset Size Description

00h 2 BYTES ???

02h WORD ???

04h WORD ???

06h 4 BYTES ???

1F1h 7 BYTES ???

-----c-137B01-----

INT 13 - NOW! v3.05 - ???

AX = 7B01h

Return: DX = segment of ???

SeeAlso: AX=7B00h

-----c-137B02-----

```
INT 13 - NOW! v3.05 - SET INFORMATION
  AX = 7B02h
  BX = segment of ??? (10h above a PSP)
  CX:DX -> 1F8h-byte information record (see #00288)
Return: ???
Program: NOW! is a disk cache by Vertisoft Systems, Inc.
Note: NOW! grabs the INT 24h value from the PSP reached via the segment in
  BX
SeeAlso: AX=7B00h
-----c-137B03-----
INT 13 - NOW! v3.05 - ???
  AX = 7B03h
  ???
Return: ???
SeeAlso: AX=7B00h,AX=7B04h
-----c-137B04-----
INT 13 - NOW! v3.05 - ???
  AX = 7B04h
  ???
Return: ???
SeeAlso: AX=7B03h
-----c-137B05-----
INT 13 - NOW! v3.05 - GET DISK ACCESSES???
  AX = 7B05h
Return: BX:AX = number of physical accesses???
  DX:CX = total disk accesses???
SeeAlso: AX=7B00h,AX=7B06h
-----c-137B06-----
INT 13 - NOW! v3.05 - GET ???
  AX = 7B06h
  BX = ???
Return: AX = 0000h
  BX = ???
SeeAlso: AX=7B05h,AX=7B07h
-----c-137B07-----
INT 13 - NOW! v3.05 - GET ???
  AX = 7B07h
Return: AX = ???
  BX = ???
  CX = ???
  DX = ???
```


SeeAlso: AX=7B06h

-----c-137B08-----

INT 13 - NOW! v3.05 - ???

AX = 7B08h

CX = ??? (default 00h)

Return: ???

SeeAlso: AX=7B00h

-----c-1380--CX6572-----

INT 13 - FAST! v4.02+ - API

AH = 80h

CX = 6572h

DX = 1970h

ES:BX -> request packet (see #00290)

AL = function number (see #00289)

Return: AH = status (except function 06h)

00h if successful

01h invalid function

05h not supported by the installed variant

CF clear if successful

CF set on error

AL may be destroyed

Program: FAST! is a disk cache by Future Computing Systems and marketed by
BLOC Publishing Corp.

SeeAlso: AX=8001h,AX=8006h,AX=8007h

Index: hotkeys;FAST!

(Table 00289)

Values for FAST! function:

01h get cache information (see AX=8001h)

04h disable cache

05h enable cache and reset statistics

06h installation check (see AX=8006h)

07h unhook interrupts (see AX=8007h)

09h flush cache

0Ah (v4.02+) enable staged writes

0Bh (v4.02+) disable staged writes

0Ch (v4.02+) enable beep on flush

0Dh (v4.02+) disable beep on flush

0Eh ???

0Fh ???

10h (v4.12+) enable hotkeys

```

11h (v4.12+) disable hotkeys
12h (v4.13+) set idle delay
13h (v4.13+) set flush dirty percentage
14h (v5.00+) enable mouse checks
15h (v5.00+) disable mouse checks
16h (v5.00d+) reduce cache size to minimum
17h (v5.00d+) increase cache size to maximum

```

Format of FAST! request packet:

Offset Size Description (Table 00290)

```

00h DWORD pointer to 19-byte signature string (see #00291)
04h DWORD pointer to buffer for data (if needed by function)

```

(Table 00291)

Values for FAST! v4.04-v5.03 signature string:

```

13h 07h 06h 08h 11h 18h 0Fh 0Eh 02h 18h 13h 08h 0Bh 08h 01h 00h 04h 08h 15h
-----c-138001CX6572-----

```

INT 13 - FAST! v4.02+ - GET CACHE INFORMATION

AX = 8001h

CX = 6572h

DX = 1970h

ES:BX -> request packet (see #00292)

Return: AH = 00h if successful

SeeAlso: AH=80h,AX=8006h

Format of FAST! request packet:

Offset Size Description (Table 00292)

```

00h DWORD -> 19-byte signature string (see #00291)
04h DWORD -> buffer for cache information (see #00293)

```

Format of FAST! cache information (v5.00-5.03):

Offset Size Description (Table 00293)

```

00h WORD binary version number of FAST! (v5.00 = 01F4h)
02h BYTE revision letter (61h = X.XXa, 62h = X.XXb, etc.)
03h BYTE FAST! variant
    (01h = FASTE, 02h = FASTX BIOS, 04h = FASTC, 20h = FASTX XMS)
04h DWORD total number of read requests
08h DWORD number of physical disk reads
0Ch DWORD grabbed hash buckets
10h DWORD "st_386mem"
14h DWORD total number of writes (only counted when staging enabled)

```

```

18h  DWORD number of physical disk writes (only when staging enabled)
1Ch  DWORD number of write errors while flushing cache
20h  WORD  flags1 (see #00294)
22h  WORD  flags
      bit 0: ???
      bit 1: staged writes enabled
24h  WORD  ???
26h  WORD  maximum cache size in KB
28h  WORD  minimum cache size in KB
2Ah  WORD  segment of first cache buffer (FASTC)
      segment of EMS page frame (FASTE)
      XMS handle (FASTX XMS)
2Ch  WORD  number of hash buckets containing no entries
2Eh  WORD  number of hash buckets containing one entry
30h  WORD  number of hash buckets containing two entries
32h  WORD  number of hash buckets containing three entries
34h  WORD  number of hash buckets containing four entries
36h  WORD  number of hash buckets containing five entries
38h  WORD  maximum contiguous sectors
3Ah  WORD  hash factor
3Ch  WORD  number of paragraphs of memory used below 1M
3Eh  WORD  entries per hash bucket
40h  WORD  idle delay in seconds
42h  2 BYTES ???
44h  WORD  staged write threshold percentage
46h  2 BYTES ???
48h  WORD  number of dirty sectors
4Ah  WORD  number of staged write buffers
4Ch  WORD  current cache size in KB
4Eh  WORD  beep frequency in Hz
50h  WORD  ???
52h  WORD  ???

```

Bitfields for FAST! flags1:

Bit(s) Description (Table 00294)

```

0  beep on flush
3  hotkeys enabled
4  mouse idle check enabled
8  caching enabled
13 ???

```

-----c-138006CX6572-----

INT 13 - FAST! v4.02+ - INSTALLATION CHECK

AX = 8006h

CX = 6572h

DX = 1970h

ES:BX -> request packet (see #00295)

Return: AX = 1965h if installed

SeeAlso: AH=80h,AX=8001h,AX=8007h

Format of FAST! request packet:

Offset Size Description (Table 00295)

00h DWORD -> 19-byte signature string (see #00291)

-----c-138007CX6572-----

INT 13 - FAST! v4.02+ - UNHOOK INTERRUPTS

AX = 8007h

CX = 6572h

DX = 1970h

ES:BX -> request packet (see #00296)

Return: AX = 1965h if installed

SeeAlso: AH=80h,AX=8006h

Index: uninstall;FAST!

Format of FAST! request packet:

Offset Size Description (Table 00296)

00h DWORD -> 19-byte signature string (see #00291)

-----c-1381--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = 81h

SI = 4358h

???

Return: ???

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of Super PC-Kwik, and thus support this call (PC-Cache v5.1 corresponds to PC-Kwik v3.20 and PC-Cache v5.5 to PC-Kwik v3.27)

returns immediately in PC-Cache v5.x

Index: PC-Cache|Qualitas Qcache

-----c-1382--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = 82h

SI = 4358h

???

Return: AL = ???

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of

Super PC-Kwik, and thus support this call

SeeAlso: AH=84h

Index: PC-Cache|Qualitas Qcache

-----c-1383--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = 83h

SI = 4358h

AL = ???

ES:BX -> ???

???

Return: ???

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of

Super PC-Kwik, and thus support this call

SeeAlso: AH=85h

Index: PC-Cache|Qualitas Qcache

-----c-1384--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = 84h

SI = 4358h

AL = ???

???

Return: AL = ???

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of

Super PC-Kwik, and thus support this call

SeeAlso: AH=82h

Index: PC-Cache|Qualitas Qcache

-----c-1385--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = 85h

SI = 4358h

AL = ???

DL = ???

???

Return: ???

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of

Super PC-Kwik, and thus support this call (PC-Cache v5.1 corresponds
to PC-Kwik v3.20)

SeeAlso: AH=83h

Index: PC-Cache|Qualitas Qcache

-----c-1386--SI4358-----

INT 13 - Super PC-Kwik v4.00+ - ???

AH = 86h

SI = 4358h

???

Return: ???

Note: Qualitas Qcache v4.00 is an OEM version of Super PC-Kwik v4.00, and
thus supports this call

Index: Qualitas Qcache

-----c-1387--SI4358-----

INT 13 - Super PC-Kwik v4.00+ - ???

AH = 87h

SI = 4358h

???

Return: AH = status??? (00h)

CX = ???

DX = ??? (0000h)

Note: Qualitas Qcache v4.00 is an OEM version of Super PC-Kwik v4.00, and
thus supports this call

Index: Qualitas Qcache

-----c-1388--SI4358-----

INT 13 - Super PC-Kwik v4.00+ - ???

AH = 88h

SI = 4358h

???

Return: AH = status??? (00h)

CX = ???

DX = ??? (0000h)

Note: Qualitas Qcache v4.00 is an OEM version of Super PC-Kwik v4.00, and
thus supports this call

Index: Qualitas Qcache

-----c-1389--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 89h

SI = 4358h

???

Return: ???

-----c-138A--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 8Ah

SI = 4358h

???

```
Return: ???
-----c-138EED-----
INT 13 - HyperDisk v4.01+ - ???
  AX = 8EEDh
  ???
Return: ???
Program: HyperDisk is a shareware disk cache by HyperWare (Roger Cross)
SeeAlso: AX=8EEEh,AX=8EEFh,AH=EEh,INT 2F/AX=DF00h
-----c-138EEE-----
INT 13 - HyperDisk v4.01+ - ???
  AX = 8EEEh
Return: CF set
  AX = CS of HyperDisk resident code
  ???
Note: identical to AX=8EEFh in HYPERDKX v4.21-4.30
SeeAlso: AX=8EEDh,AX=8EEFh,AH=EEh
-----c-138EEF-----
INT 13 - HyperDisk v4.01+ - ???
  AX = 8EEFh
Return: CF set
  AX = CS of HyperDisk resident code
  ???
Note: identical to AX=8EEEh in HYPERDKX v4.21-4.30
SeeAlso: AX=8EEDh,AX=8EEEh,AH=EEh
-----c-1392--SI4358-----
INT 13 - Super PC-Kwik v5.10+ - ???
  AH = 92h
  SI = 4358h
  ???
Return: AH = status??? (00h)
  DL = ???
SeeAlso: AH=93h
-----c-1393--SI4358-----
INT 13 - Super PC-Kwik v5.10+ - ???
  AH = 93h
  SI = 4358h
  ???
Return: AH = status??? (00h)
  AL = ???
SeeAlso: AH=92h
-----c-1394--SI4358-----
```

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 94h

SI = 4358h

???

Return: ???

-----c-1395--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 95h

SI = 4358h

???

Return: AH = status??? (00h)

DX = ???

-----c-1396--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 96h

SI = 4358h

AL = ??? (01h)

BX = ??? (0790h)

DL = ???

Return: AH = status??? (00h)

DX = ???

-----c-1397--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 97h

SI = 4358h

???

Return: ???

-----c-1398--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 98h

SI = 4358h

???

Return: ???

-----c-1399--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???

AH = 99h

SI = 4358h

???

Return: ???

-----c-139A--SI4358-----

INT 13 - Super PC-Kwik v5.10+ - ???


```
AH = 9Ah
SI = 4358h
???
```

```
Return: ???
```

```
-----c-139B--SI4358-----
```

```
INT 13 - Super PC-Kwik v5.10+ - ???
```

```
AH = 9Bh
SI = 4358h
???
```

```
Return: ???
```

```
-----c-139C--SI4358-----
```

```
INT 13 - Super PC-Kwik v5.10+ - ???
```

```
AH = 9Ch
SI = 4358h
???
```

```
Return: ???
```

Note: functions 9Ch and 9Dh are the only ones which are fully reentrant; all other PC-Kwik API calls (INT 13/81h-B0h) return AX=0200h and CF clear if a previous call is still in progress

```
-----c-139D--SI4358-----
```

```
INT 13 - Super PC-Kwik v5.10+ - ???
```

```
AH = 9Dh
SI = 4358h
???
```

```
Return: ???
```

```
-----c-13A0--SI4358-----
```

```
INT 13 - Super PC-Kwik v3.20+ - GET RESIDENT CODE SEGMENT
```

```
AH = A0h
SI = 4358h
```

```
Return: AX = segment of resident code
```

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of Super PC-Kwik, and thus support this call (note that PC-Cache v5.5 corresponds to PC-Kwik v3.27)

SeeAlso: INT 16/AX=FFA5h/CX=1111h

Index: PC-Cache|Qualitas Qcache

```
-----c-13A1--SI4358-----
```

```
INT 13 - Super PC-Kwik v3.20+ - FLUSH CACHE
```

```
AH = A1h
SI = 4358h
```

```
Return: CF clear
```

```
AH = 00h (v5.10)
```

Notes: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of

Super PC-Kwik, and thus support this call (note that PC-Cache v5.1 corresponds to PC-Kwik v3.20)

this function is one out of several cache flush calls issued by the

PTS-DOS 6.51 and S/DOS 1.0 kernel before rebooting.

SeeAlso: INT 16/AX=FFA5h/CX=FFFFh

Index: PC-Cache|Qualitas Qcache

-----c-13A2--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = A2h

SI = 4358h

???

Return: ???

Note: PC Tools PC-Cache 5.x and Qualitas Qcache 4.00 are OEM versions of

Super PC-Kwik, and thus support this call (note that PC-Cache v5.1 corresponds to PC-Kwik v3.20)

Index: PC-Cache|Qualitas Qcache

-----c-13A3--SI4358-----

INT 13 U - Super PC-Kwik v5.10+ - DISABLE CACHE

AH = A3h

SI = 4358h

Return: CF clear

SeeAlso: AH=A4h

-----c-13A4--SI4358-----

INT 13 U - Super PC-Kwik v5.10+ - ENABLE CACHE

AH = A4h

SI = 4358h

Return: CF clear

SeeAlso: AH=A3h

-----c-13A5--SI4358-----

INT 13 CU - Super PC-Kwik v5.10+ - PROGRAM TERMINATION NOTIFICATION

AH = A5h

SI = 4358h

Return: AX = ???

SI = ???

Notes: called and used internally by Super PC-Kwik when a program terminates

via INT 21/AH=00h, INT 21/AH=31h, or INT 21/AH=4Ch

this call is not supported by Qualitas Qcache 4.00

Index: PC-Cache

SeeAlso: AH=A6h,AH=A9h,INT 21/AH=00h,INT 21/AH=31h,INT 21/AH=4Ch

-----c-13A6--SI4358-----

INT 13 CU - Super PC-Kwik v5.10+ - PROGRAM LOAD NOTIFICATION

AH = A6h

SI = 4358h

DS:DX -> ASCIZ program name

ES:BX -> EXEC data block (see #01590 at INT 21/AH=4Bh)

Return: ???

Note: called and used internally by Super PC-Kwik when a program is loaded
with INT 21/AX=4B00h

SeeAlso: AH=A5h,AH=A9h,INT 21/AH=4Bh

-----c-13A7--SI4358-----

INT 13 CU - Super PC-Kwik 5.1 - ???

AH = A7h

SI = 4358h

Return: ???

Note: called and used internally by Super PC-Kwik on some INT 21 calls

SeeAlso: AH=A5h,AH=A6h,AH=A8h

-----v-13A759-----

INT 13 U - Novell DOS 7 - SDRes v27.03 - ???

AX = A759h

Return: AX = 59A7h if installed

DX:BX -> ??? data

Program: SDRes is the resident portion of the Search&Destroy antiviral by
Fifth Generation Systems, as bundled with Novell DOS 7

SeeAlso: INT 21/AH=0Eh/DL=ADh

-----c-13A8--SI4358-----

INT 13 CU - Super PC-Kwik 5.1 - ???

AH = A8h

SI = 4358h

Return: ???

Note: called and used internally by Super PC-Kwik on some INT 21 calls

SeeAlso: AH=A5h,AH=A6h,AH=A7h

-----c-13A9--SI4358-----

INT 13 CU - Super PC-Kwik 5.1 - EXITCODE RETRIEVAL NOTIFICATION

AH = A9h

SI = 4358h

Return: ???

Note: called and used internally by Super PC-Kwik when an application issues
INT 21/AH=4Dh

SeeAlso: AH=A5h,AH=A6h,INT 21/AH=4Dh

-----c-13AA--SI4358-----

INT 13 - Super PC-Kwik v4+ - ???

```
AH = AAh
SI = 4358h
???
```

Return: ???

Note: Qualitas Qcache is an OEM version of Super PC-Kwik, and thus supports this call

```
-----c-13AB--SI4358-----
```

INT 13 - Super PC-Kwik v4+ - ???

```
AH = ABh
SI = 4358h
???
```

Return: ???

Note: Qualitas Qcache is an OEM version of Super PC-Kwik, and thus supports this call

```
-----c-13AC--SI4358-----
```

INT 13 - Super PC-Kwik v4+ - ???

```
AH = ACh
SI = 4358h
???
```

Return: ???

Note: Qualitas Qcache is an OEM version of Super PC-Kwik, and thus supports this call

```
-----d-13AD-----
```

INT 13 - Priam HARD DISK CONTROLLER???

```
AH = ADh
???
```

Return: ???

Note: this call is made from Priam's EFMT.EXE (low-level formatter), probably to check the ROM type on the controller for their hard disk kits

SeeAlso: AH=70h

```
-----c-13AD--SI4358-----
```

INT 13 - Super PC-Kwik v4+ - ???

```
AH = ADh
SI = 4358h
???
```

Return: ???

Note: Qualitas Qcache is an OEM version of Super PC-Kwik, and thus supports this call

```
-----c-13AE--SI4358-----
```

INT 13 - Super PC-Kwik v5.10+ - ???

```
AH = AEh
```

SI = 4358h

???

Return: ???

-----c-13B0--SI4358-----

INT 13 - Super PC-Kwik v3.20+ - ???

AH = B0h

SI = 4358h

???

Return: ???

Note: PC Tools PC-Cache 5.x is an OEM version of Super PC-Kwik, and thus supports this call; Qualitas Qcache does not support it

Index: PC-Cache

-----13BF00-----

INT 13 - Mylex/Adaptec??? - ???

AX = BF00h

BX = PCI address???

DI = 0000h ???

DL = drive number (80h = C:)

Return: ???

Note: some poorly-commented code using this function notes that there had been a workaround for some Mylex BIOS bug by setting DL to 8Fh

-----d-13E000-----

INT 13 - XBIOS - COMMAND

AX = E000h

CX = 0

DL = drive number (80h,81h,82h,83h)

ES:BX = pointer to XBIOS Command Block (see #00297)

Return: CF clear if successful

CX = 1234h XBIOS Signature

CF set on error

Program: XBIOS is a driver in some versions of Disk Manager that is loaded from the disk MBR, replacing the ROM BIOS disk support e.g. adding LBA mode support, and read/write multiple.

SeeAlso: AH=F9h"SWBIOS"

Format of XBIOS Command Block:

Offset Size Description (Table 00297)

00h BYTE function

0Eh Get XBIOS Configuration Information

Return: buffer points to Ontrackr_Ref_Data structure

(see #00298)

01h BYTE reserved (must be zero before function call)
 02h DWORD buffer pointer - Input or output depending on opcode
 SeeAlso: #00298,#00299

Format of Ontrackr_Ref_Data structure:

Offset	Size	Description (Table 00298)
00h	WORD	Size of structure (33h)
02h	BYTE	VxD Chain mode
03h	12 BYTES	Drive 1: Ontrackr_VxD_Data structure (see #00299)
0Fh	12 BYTES	Drive 2: Ontrackr_VxD_Data structure
1Bh	12 BYTES	Drive 3: Ontrackr_VxD_Data structure
27h	12 BYTES	Drive 4: Ontrackr_VxD_Data structure (see #00299)

SeeAlso: #00297,#00299

Format of Ontrackr_VxD_Data structure:

Offset	Size	Description (Table 00299)
00h	BYTE	INT 13h drive number
01h	DWORD	Delta skew value
05h	BYTE	Physical heads (Word 3 of Identify Data)
06h	BYTE	Physical sectors per track (Word 6 of Identify Data)
07h	BYTE	Multiple Block Size (Blocking factor)
08h	BYTE	Read/Write Multiple disable flags
		00001000b drive supports r/w multiple
		00000100b do not use read multiple
		00000010b do not use write multiple
09h	WORD	Base port address (1F0/170)
0Bh	BYTE	Hardware interrupt channel (14/15)

SeeAlso: #00298

-----v-13EC00-----

INT 13 - VIRUS - "Tiso" - INSTALLATION CHECK

AX = EC00h

Return: CF clear if installed

SeeAlso: AH=F2h, INT 12/AX=4350h/BX=4920h

-----d-13EE-----

INT 13 - SWBIOS - SET 1024-CYLINDER FLAG

AH = EEh

DL = drive number (80h, 81h)

Return: CF clear

AH = 00h

Program: SWBIOS is a TSR by Ontrack Computer Systems

Desc: the following INT 13 call will add 1024 to the specified cylinder

number to get the actual cylinder number desired

Notes: the flag is cleared by all INT 13 calls except AH=EEh and AH=EFh

Disk Manager also supports these calls

this function is also supported by HyperDisk v4.01+ and PC-Cache v5.5+,
in order to allow caching of drives using SWBIOS to access more than
1024 cylinders

for software which supports that call, this function is equivalent to
calling AH=EFh with CX=0400h

SeeAlso: AH=F9h,AH=FEh,INT 16/AX=FFA5h/CX=1111h,INT 2F/AX=DF00h

Index: PC-Cache;huge disks|Disk Manager

-----c-13EF-----

INT 13 - Ontrack Drive Rocket - SET CYLINDER OFFSET

AH = EFh

CX = cylinder offset for next INT 13 call

DL = drive number (80h, 81h)

Return: CF clear

AH = 00h

Program: Drive Rocket is a drive accelerator by Ontrack Computer Systems for
IDE drives supporting the read multiple and write multiple commands

Desc: the following INT 13 call will add the number given by this call to
the specified cylinder to get the actual cylinder number, then reset
the offset to zero

Note: this function is also supported by the NOW! disk cache, and presumably
newer versions of SWBIOS and Disk Manager

for software which supports this call, AH=EEh is equivalent to calling
this function with CX=0400h

the cylinder offset is reset to 0 by all INT 13 called except AH=EEh
and AH=EFh

SeeAlso: AX=7B00h

-----v-13F2-----

INT 13 - VIRUS - "Neuroquila" - INSTALLATION CHECK

AH = F2h

Return: CF ??? if installed

SeeAlso: AX=EC00h,INT 12/AX=4350h/BX=4920h,INT 21/AX=0B56h

-----d-13F9-----

INT 13 - SWBIOS - INSTALLATION CHECK

AH = F9h

DL = drive number (80h,81h)

Return: CF clear

DX = configuration word

bit 15 set if other SWBIOS extensions available

CF set on error

Program: SWBIOS is a TSR by Ontrack Computer Systems

XBIOS is a driver in some versions of Disk Manager that is loaded from the disk MBR, replacing the ROM BIOS disk support eg adding LBA mode support, and read/write multiple.

Note: Disk Manager also supports these calls

SeeAlso: AH=EEh,AX=E000h"XBIOS",AH=FFh"EZ-Drive"

Index: Disk Manager

-----v-13FA--DX5945-----

INT 13 - PC Tools v8+ VSAFE, VWATCH - API

AH = FAh

DX = 5945h

AL = function (00h-07h)

Return: varies by function

if not installed:

CF set

AH = 01h

Note: this API is identical to the ones on INT 16/AH=FAh and INT 21/AH=FAh, so it is listed in its entirety under INT 16/AX=FA00h and following

SeeAlso: INT 16/AX=FA00h

-----v-13FD50-----

INT 13 - VIRUS - "Predator" - INSTALLATION CHECK

AX = FD50h

Return: AX = 50FDh if resident

SeeAlso: AX=5001h"VIRUS",INT 16/AH=DDh"VIRUS"

-----d-13FE-----

INT 13 - SWBIOS - GET EXTENDED CYLINDER COUNT

AH = FEh

DL = drive number (80h, 81h)

Return: CF clear

DX = number of cylinders beyond 1024 on drive

Program: SWBIOS is a TSR by Ontrack Computer Systems

Notes: standard INT 13/AH=08h will return a cylinder count truncated to 1024

BIOS without this extension would return count modulo 1024

Disk Manager also supports these calls

SeeAlso: AH=EEh

-----13FF-----

INT 13 - EZ-Drive - INSTALLATION CHECK

AH = FFh

DL = drive number (80h)

Return: CF clear

AX = AA55h

ES:BX -> string "AERMH13Vxx", where xx is the version number of
the EZ-Drive driver

CF set on error

Program: EZ-Drive is a driver by Micro House that is loaded from the
hard disk MBR, replacing the ROM BIOS disk support, eg adding
LBA mode support, and read/write multiple.

Note: this function is called by the Windows95 Master Boot Record

SeeAlso: AX=E000h"XBIOS",AH=F9h"SWBIOS"

-----B-13FF-----

INT 13 - IBM SurePath BIOS - Officially "Private" Function

AH = FFh

-----U-13FFFFBHAA-----

INT 13 - UNIQUE UX Turbo Utility - SET TURBO MODE

AX = FFFFh

BH = AAh

BL = subfunction

00h installation check

Return: AX = 1234h if installed

01h turn on Turbo mode

02h turn off Turbo mode

03h set Turbo mode according to hardware switch

04h set disk access to Turbo mode

05h set disk access to Normal mode

Return: nothing

SeeAlso: INT 15/AH=DFh

Index: installation check;UNIQUE UX Turbo Utility

-----S-14-----

INT 14 - SERIAL - Digiboard DigiCHANNEL PC/X* Extender INT 14 (XAPCM232.SYS)

InstallCheck: determine whether the "~DOSXAM~" character device exists

Index: installation check;Digiboard DigiCHANNEL

-----S-1400-----

INT 14 - SERIAL - INITIALIZE PORT

AH = 00h

AL = port parameters (see #00300)

DX = port number (00h-03h) (04h-43h for Digiboard XAPCM232.SYS)

Return: AH = line status (see #00304)

FFh if error on Digiboard XAPCM232.SYS

AL = modem status (see #00305)

Notes: default handler is at F000h:E739h in IBM PC and 100% compatible BIOSes

since the PCjr supports a maximum of 4800 bps, attempting to set 9600

bps will result in 4800 bps
 various network and serial-port drivers support the standard BIOS
 functions with interrupt-driven I/O instead of the BIOS's polled I/O
 the 1993/04/08 Compaq system ROM uses only the low two bits of DX
 the default setting used by DOS (MS-DOS 6, DR-DOS 7.03, PTS-DOS) when
 (re-)initializing the serial devices is AL=A3h (2400 bps, no parity,
 1 stop bit, 8 data bits).
 SeeAlso: AH=04h"SERIAL",AH=04h"MultiDOS",AH=05h"SERIAL",AH=57h
 SeeAlso: AX=8000h"ARTICOM",AH=81h"COMM-DRV",AH=82h"COURIERS",AH=8Ch
 SeeAlso: MEM 0040h:0000h,PORT 03F8h"Serial"

Bitfields for serial port parameters:

Bit(s) Description (Table 00300)

7-5 data rate (110,150,300,600,1200,2400,4800,9600 bps)
 4-3 parity (00 or 10 = none, 01 = odd, 11 = even)
 2 stop bits (set = 2, clear = 1)
 1-0 data bits (00 = 5, 01 = 6, 10 = 7, 11 = 8)

SeeAlso: #00302,#00307,#00308,#00309

-----S-1400-----

INT 14 - FOSSIL (Fido/Opus/Seadog Standard Interface Level) - INITIALIZE

AH = 00h

AL = initializing parameters

7	6	5	4	3	2	1	0
-BAUD RATE-			PARITY		STOP		WORD
		BITS	LENGTH				
000	19200	bd	00 none	0:	1	00:	5
001	38400	bd	01 odd	1:	2	01:	6
010	300	bd	11 even	10:			7
011	600	bd		11:			8
100	1200	bd					
101	2400	bd					
110	4800	bd					
111	9600	bd	(4800 on PCjr)				

DX = port number (0-3 or FFh if only performing non-I/O setup)

Return: AH = RS-232 status code bits (see #00301)

AL = modem status bits

bit 3: always 1

bit 7: DCD - carrier detect

SeeAlso: #00300,AH=05h"FOSSIL",AH=81h"COMM-DRV",AH=82h"COURIERS"

Bitfields for FOSSIL RS-232 status:

Bit(s) Description (Table 00301)

- 0 RDA - input data is available in buffer
- 1 OVRN - data has been lost
- 5 THRE - room is available in output buffer
- 6 TSRE - output buffer empty

-----S-1400-----
INT 14 - Tandy 2000 - SERIAL - RESET COMM PORT

- AH = 00h
- AL = RS-232C parameters (see #00302)
- DL = port number
- DH = protocol
 - bit 0: use XON/XOFF on received data
 - bit 1: use XON/XOFF when transmitting

Return: AH = line status (see #00304)

- AL = modem status (see #00305)

Note: this interrupt is identical to INT 53 on the Tandy 2000

SeeAlso: AH=04h"Tandy 2000",INT 53"Tandy 2000"

-----S-1400-----
INT 14 - MBBIOS - INITIALIZE PORT

- AH = 00h
- AL = port parameters (see #00302)
- DX = port number

Return: AH = line status (see #00304)

- AL = modem status (see #00305)

Note: MBBIOS was written by H. Roy Engehausen

SeeAlso: AH=04h"MBBIOS",AH=05h"MBBIOS",AH=09h"MBBIOS"

Bitfields for MBBIOS port parameters:

Bit(s) Description (Table 00302)

- 7-5 data rate
 - (normally 110,150,300,600,1200,2400,4800,9600 bps;
 - 9600,14400,19200,28800,38400,57600,115200,330400 bps
 - if the high-speed option is set)
- 4-3 parity (00 or 10 = none, 01 = odd, 11 = even)
- 2 stop bits (set = 2, clear = 1)
- 1-0 data bits (00 = 5, 01 = 6, 10 = 7, 11 = 8)

SeeAlso: #00300

-----N-1400--DXFFFF-----
INT 14 - Connection Manager - MODIFY DEFAULT CONNECTION PARAMETERS

- AH = 00h
- DX = FFFFh

ES:DI -> vector string specifying new parameters
Return: AH = return code (00h,03h) (see #00303)
Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network
Note: if DX is 0-3 on entry, Connection Manager emulates the standard BIOS
function, but redirects the port over the network; if DX is any other
value, the call is chained
SeeAlso: AH=04h/DX=FFFFh,AH=08h/DX=FFFFh,AH=0Ah/DX=FFFFh

(Table 00303)

Values for Connection Manager return code:

00h successful
01h no such connection
02h invalid connection ID
03h invalid subvector found
04h communication error (check BH)
06h insufficient resources, retry later
FFh no data available

-----S-1401-----

INT 14 - SERIAL - WRITE CHARACTER TO PORT

AH = 01h
AL = character to write
DX = port number (00h-03h) (04h-43h for Digiboard XAPCM232.SYS)

Return: AH bit 7 clear if successful

AH bit 7 set on error
AH bits 6-0 = port status (see #00304)

Notes: various network and serial-port drivers support the standard BIOS
functions with interrupt-driven I/O instead of the BIOS's polled I/O
the 1993/04/08 Compaq system ROM uses only the low two bits of DX

SeeAlso: AH=02h,AH=0Bh"FOSSIL",AX=8000h"ARTICOM",AH=89h,MEM 0040h:007Ch

-----N-1401--DXFFFF-----

INT 14 - Connection Manager - SEND CHARACTER

AH = 01h
DX = FFFFh
BH = character to send

Return: AH = return code (00h-02h,06h) (see #00303)

Notes: if DX is 0-3 on entry, Connection Manager emulates the standard BIOS
function, but redirects the port over the network; if DX is any other
value, the call is chained

this function is provided primarily for compatibility; AH=06h/DX=FFFFh
is the preferred function because it provides better performance

SeeAlso: AH=02h/DX=FFFFh,AH=06h/DX=FFFFh,AH=09h/DX=FFFFh

-----S-1402-----

INT 14 - SERIAL - READ CHARACTER FROM PORT

AH = 02h

AL = 00h (ArtiCom)

DX = port number (00h-03h (04h-43h for Digiboard XAPCM232.SYS))

Return: AH = line status (see #00304)

AL = received character if AH bit 7 clear

Notes: will timeout if DSR is not asserted, even if function 03h returns
data ready

various network and serial-port drivers support the standard BIOS
functions with interrupt-driven I/O instead of the BIOS's polled I/O
the 1993/04/08 Compaq system ROM uses only the low two bits of DX

SeeAlso: AH=01h,AH=02h"FOSSIL",AH=84h,AH=FCh

-----S-1402-----

INT 14 - FOSSIL - RECEIVE CHARACTER WITH WAIT

AH = 02h

DX = port number (0-3)

Return: AL = character received

AH = 00h

SeeAlso: AH=01h,AH=02h"SERIAL"

-----N-1402--DXFFFF-----

INT 14 - Connection Manager - RECEIVE CHARACTER

AH = 02h

DX = FFFFh

BH = character to send

Return: AH = return code (00h-02h,04h,FFh) (see #00303)

BH = line status (see #00304)

AL = received character (if any)

Notes: if DX is 0-3 on entry, Connection Manager emulates the standard BIOS
function, but redirects the port over the network; if DX is any other
value, the call is chained

this function is provided primarily for compatibility; AH=07h/DX=FFFFh
is the preferred function because it provides better performance

SeeAlso: AH=02h/DX=FFFFh,AH=03h/DX=FFFFh,AH=06h/DX=FFFFh

-----S-1403-----

INT 14 - SERIAL - GET PORT STATUS

AH = 03h

AL = 00h (ArtiCom)

DX = port number (00h-03h) (04h-43h for Digiboard XAPCM232.SYS)

Return: AH = line status (see #00304)

AL = modem status (see #00305)

AX = 9E00h if disconnected (ArtiCom)

Note: the 1993/04/08 Compaq system ROM uses only the low two bits of DX

SeeAlso: AH=00h,AH=07h"MultiDOS",AX=8000h"ARTICOM",AH=81h"COURIERS",AX=FD02h

Bitfields for serial line status:

Bit(s) Description (Table 00304)

7 timeout
6 transmit shift register empty
5 transmit holding register empty
4 break detected
3 framing error
2 parity error
1 overrun error
0 receive data ready

Note: for COMM-DRV, if bit 7 is set, an error occurred, and may be retrieved through a separate call (see AX=8000h"COMM-DRV")

Bitfields for modem status:

Bit(s) Description (Table 00305)

7 carrier detect
6 ring indicator
5 data set ready
4 clear to send
3 delta carrier detect
2 trailing edge of ring indicator
1 delta data set ready
0 delta clear to send

-----N-1403--DXFFFF-----

INT 14 - Connection Manager - RETURN COMMUNICATION PORT STATUS

AH = 03h

DX = FFFFh

AL = connection ID

Return: AH = return code (00h-02h) (see #00303)

BH = line status (see #00306)

BL = modem status (see #00305) (only bits 4,5,7; all others zero)

Notes: if DX is 0-3 on entry, Connection Manager emulates the standard BIOS function, but redirects the port over the network; if DX is any other value, the call is chained

SeeAlso: AH=00h/DX=FFFFh,AH=04h/DX=FFFFh,AH=0Ah/DX=FFFFh

Bitfields for Connection Manager line status:

Bit(s) Description (Table 00306)

7 CTS changed
6 current CTS state
5 timeout
4 break
3 framing error
2 parity error
1 overrun
0 current carrier state (0 active, 1 no carrier)

-----S-1404-----

INT 14 - SERIAL - EXTENDED INITIALIZE (CONVERTIBLE,PS)

AH = 04h

AL = break status

00h if break

01h if no break

BH = parity (see #00307)

BL = number of stop bits

00h one stop bit

01h two stop bits (1.5 if 5 bit word length)

CH = word length (see #00308)

CL = bps rate (see #00309)

DX = port number

Return: AX = port status code (see #00304,#00305)

SeeAlso: AH=00h,AH=1Eh,AX=8000h"ARTICOM"

(Table 00307)

Values for serial port parity:

00h no parity

01h odd parity

02h even parity

03h stick parity odd

04h stick parity even

SeeAlso: #00300,#00308,#00309,#00310

(Table 00308)

Values for serial port word length:

00h 5 bits

01h 6 bits

02h 7 bits

03h 8 bits

SeeAlso: #00300,#00307,#00309,#00345

(Table 00309)

Values for serial port bps rate:

00h 110 (19200 if ComShare installed)

01h 150 (38400 if ComShare installed)

02h 300

03h 600 (14400 if ComShare installed)

04h 1200

05h 2400

06h 4800 (28800 if ComShare installed)

07h 9600

08h 19200

---ComShare---

09h 38400

0Ah 57600

0Bh 115200

SeeAlso: #00300,#00307,#00309,#00346,#00353,AH=36h,#00364,#00606,#02923

-----S-1404-----

INT 14 - Tandy 2000 - SERIAL - FLUSH COMM BUFFER

AH = 04h

DL = port number

DH = protocol

bit 0: use XON/XOFF on received data

bit 1: use XON/XOFF when transmitting

Return: nothing

Desc: clears the serial interface buffer

Note: this interrupt is identical to INT 53 on the Tandy 2000

SeeAlso: AH=00h"Tandy 2000",INT 53"Tandy 2000"

-----S-1404-----

INT 14 - FOSSIL - INITIALIZE DRIVER

AH = 04h

DX = port number

optionally BX=4F50h

ES:CX -> byte to be set upon ^C

Return: AX = 1954h (if successful)

BL = maximum function number supported (excluding 7Eh and above)

BH = revision of FOSSIL specification supported

DTR is raised

Note: the word at offset 6 in the interrupt handler contains 1954h, and the following byte contains the maximum function number supported; this

can serve as an installation check

SeeAlso: AH=05h"FOSSIL",AH=1Ch,INT 11/AH=BCh

Index: installation check;FOSSIL

-----S-1404-----

INT 14 - MultiDOS Plus IODRV - INITIALIZE PORT

AH = 04h

Return: port initialized; if Hayes-compatible modem, a connection has been established

Note: the port number is stored at offset BEh in the Task Control Block
(see #00456 at INT 15/AH=13h"MultiDOS")

SeeAlso: AH=00h,AH=05h"MultiDOS",AH=20h"MultiDOS",INT 15/AH=13h"MultiDOS"

-----S-1404-----

INT 14 - Digiboard DigiCHANNEL PC/X* - CHANGE BAUD RATE

AH = 04h

AL = initializing parameters (see #00310)

BX = baud rate

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

FFh error

SeeAlso: AH=05h"Digiboard"

Bitfields for Digiboard initializing parameters:

Bit(s) Description (Table 00310)

7-5 unused

4-3 parity (00 none, 01 odd, 11 even)

2 stop bits (0 = one, 1 = two)

1-0 data bits (00 = five, 01 = six, 10 = seven, 11 = eight)

SeeAlso: #00307,#00308

-----S-1404-----

INT 14 - MBBIOS - INSTALLATION CHECK

AH = 04h

DX = port number

Return: AX = AA55h if installed on specified port

SeeAlso: AH=00h"MBBIOS",AH=09h"MBBIOS"

-----N-1404--DXFFFF-----

INT 14 - Connection Manager - OPEN COMMUNICATION

AH = 04h

DX = FFFFh

ES:DI -> Connection Request protocol vector (see #00311)

Return: AH = return code

00h successful
 AL = connection ID
 BH = connection type
 00h direct connection or no dialing
 01h Connection Server dialed phone
 01h no response from Connection Server
 03h invalid request

Program: Connection Manager by Softwarehouse Corp. permits the sharing of serial ports over an IPX or NetBIOS-based network

Desc: initiate a connection to the Connection Server listed in the current Client parameter set

Notes: if DX is 0-3 on entry, Connection Manager emulates the standard BIOS function, but redirects the port over the network; if DX is any other value, the call is chained

all subvectors of the Connection Request vector are optional; if missing, default values are provided by the default connection parameter set

SeeAlso: AH=00h/DX=FFFFh, AH=05h/DX=FFFFh, AH=06h/DX=FFFFh, AH=07h/DX=FFFFh

SeeAlso: AH=0Ah/DX=FFFFh, AH=0Ch/DX=FFFFh

Format of Connection Manager protocol command vector:

Offset	Size	Description (Table 00311)
00h	WORD	(big-endian) total length of command (including this word)
02h	WORD	(big-endian) command code
		EF01h Connection Request
		EF06h Modify Connection Parameters
04h	N BYTES	list of subvectors (see #00313)
		allowable subvector types are 01h-04h, 17h, 18h for command code
		EF01h; 03h, 04h for command code EF06h (see #00312)

(Table 00312)

Values for Connection Manager subvector type code:

01h Connection ID
 02h Destination ID
 03h Asynchronous line parameters
 04h Data transfer parameters
 09h Line speed
 0Ah Serial coding
 0Bh Packet size
 0Ch Timers
 0Dh Special characters

0Eh Target ID
0Fh Telephone number
10h ASCII destination ID
11h Parity
12h Bits per character
13h Number of stop bits
14h Packet timer
15h Intercharacter timer
17h Flags
18h Parameter ranges
19h Flow control

Format of Connection Manager subvector:

Offset Size Description (Table 00313)

00h BYTE length of subvector
01h BYTE type code (see #00312)
02h N-2 BYTES data, which may include subvectors

SeeAlso: #00314,#00315,#00316,#00317,#00318,#00319,#00320,#00321,#00322,#00323,#00324

SeeAlso: #00325,#00326,#00328,#00329,#00330,#00331,#00332,#00333,#00311

Format of Connection ID subvector:

Offset Size Description (Table 00314)

00h BYTE 03h (length)
01h BYTE 01h (subvector "Connection ID")
02h BYTE connection ID

SeeAlso: #00313

Format of Destination ID subvector:

Offset Size Description (Table 00315)

00h BYTE length
01h BYTE 02h (subvector "Destination ID")
02h N BYTES subvector(s) of type 0Eh, 0Fh, or 10h

SeeAlso: #00313

Format of Asynchronous line parameters subvector:

Offset Size Description (Table 00316)

00h BYTE length
01h BYTE 03h (subvector "Asynchronous line parameters")
02h N BYTES subvector(s) of type 09h, 0Ah, or 19h

SeeAlso: #00313

Format of Data transfer parameters subvector:

Offset	Size	Description (Table 00317)
00h	BYTE	length
01h	BYTE	04h (subvector "Data transfer parameters")
02h	N BYTES	subvector(s) of type 0Bh, 0Ch, or 0Dh

SeeAlso: #00313

Format of Line speed subvector:

Offset	Size	Description (Table 00318)
00h	BYTE	04h (length)
01h	BYTE	09h (subvector "Line speed")
02h	WORD	bit map, highest set bit selects speed
		bit 0: 2400
		bits 1-7: 1800, 1200, 600, 300, 115200, 150, 110 bps
		bits 8-15: 57600, 38400, 19200, 14400, 9600, 7200, 4800, 3600

SeeAlso: #00313

Format of Serial coding subvector:

Offset	Size	Description (Table 00319)
00h	BYTE	length
01h	BYTE	0Ah (subvector "Serial coding")
02h	N BYTES	subvector(s) of type 11h, 12h, or 13h

SeeAlso: #00313

Format of Packet size subvector:

Offset	Size	Description (Table 00320)
00h	BYTE	04h (length)
01h	BYTE	0Bh (subvector "Packet size")
02h	WORD	(big-endian) packet size, 1 to 1024

SeeAlso: #00313

Format of Timers subvector:

Offset	Size	Description (Table 00321)
00h	BYTE	length
01h	BYTE	0Ch (subvector "Timers")
02h	8 BYTES	subvector of type 14h or 15h

SeeAlso: #00313

Format of Special characters subvector:

Offset	Size	Description (Table 00322)
00h	BYTE	length

01h BYTE 0Dh (subvector "Special characters")
02h N BYTES list of ASCII characters to be used as EOM or EOB
SeeAlso: #00313

Format of Target ID:

Offset Size Description (Table 00323)
00h BYTE length
01h BYTE 0Eh (subvector "Target ID")
02h N BYTES target ID, 1-16 bytes
SeeAlso: #00313

Format of Telephone number subvector:

Offset Size Description (Table 00324)
00h BYTE length
01h BYTE 0Fh (subvector "Telephone number")
02h N BYTES telephone number
SeeAlso: #00313

Format of ASCII destination ID subvector:

Offset Size Description (Table 00325)
00h BYTE length
01h BYTE 10h (subvector "ASCII destination ID")
02h N BYTES destination ID
SeeAlso: #00313

Format of Parity subvector:

Offset Size Description (Table 00326)
00h BYTE 03h (length)
01h BYTE 11h (subvector "Parity")
02h BYTE parity type (see #00327)
SeeAlso: #00313

Bitfields for Connection Manager parity type:

Bit(s) Description (Table 00327)
7 odd
6 even
5 mark
4 space
3 none
SeeAlso: #00326

Format of Bits per character subvector:

Offset Size Description (Table 00328)

00h BYTE 03h (length)
01h BYTE 12h (subvector "Bits per character")
02h BYTE bits per character
 bit 7: seven
 bit 6: eight

SeeAlso: #00313

Format of Number of stop bits subvector:

Offset Size Description (Table 00329)

00h BYTE 03h (length)
01h BYTE 13h (subvector "Number of stop bits")
02h BYTE stop bits
 bit 7: one
 bit 6: 1.5
 bit 5: two

SeeAlso: #00313

Format of Packet timer and Intercharacter timer subvectors:

Offset Size Description (Table 00330)

00h BYTE 04h (length)
01h BYTE subvector type
 14h Packet timer
 15h Intercharacter timer
02h WORD (big-endian) unit of value representing 20ms

SeeAlso: #00313

Format of Flags subvector:

Offset Size Description (Table 00331)

00h BYTE 03h (length)
01h BYTE 17h (subvector "Flags")
02h BYTE flags
 bit 7: queueing requested

SeeAlso: #00313

Format of Parameter ranges subvector:

Offset Size Description (Table 00332)

00h BYTE length
01h BYTE 18h (subvector "Parameter ranges")
02h N BYTES subvector(s) of type 09h, 11h, 12h, or 13h

SeeAlso: #00313

Format of Flow control subvector:

Offset Size Description (Table 00333)

00h BYTE length (02h-04h)

01h BYTE 19h (subvector "Flow control")

02h BYTE XOFF character

03h BYTE XON character

Note: if length is 02h, flow control is disabled; if length is 03h, any character will be accepted as XON after an XOFF

SeeAlso: #00313

-----S-140400-----

INT 14 - Microsoft Systems Journal TSRCOMM INT14 - INSTALLATION CHECK

AX = 0400h

Return: AX = 0FF0h

SeeAlso: AX=0401h,AX=0408h

-----S-140401-----

INT 14 - Microsoft Systems Journal TSRCOMM INT14 - INITIALIZE MODE

AX = 0401h

CX = mode

Return: nothing

SeeAlso: AX=0400h,AX=0402h

-----S-140402-----

INT 14 - Microsoft Systems Journal TSRCOMM INT14 - EXTENDED INITIALIZE

AX = 0402h

CL = parameters

Return: nothing

SeeAlso: AX=0400h,AX=0401h

-----S-140403-----

INT 14 - Microsoft Systems Journal TSRCOMM INT14 - SET TIMEOUT

AX = 0403h

CX = timeout

Return: nothing

SeeAlso: AX=0400h

-----S-140404-----

INT 14 - Microsoft Systems Journal TSRCOMM INT14 - CLEAR THE RECEIVE BUFFER

AX = 0404h

Return: nothing

SeeAlso: AX=0400h,AX=0405h,AX=0406h

-----S-140405-----

INT 14 - Microsoft Systems Journal TSRCOMM INT14 - GET RECEIVE BUFFER COUNT

AX = 0405h
Return: AX = number of characters in buffer
SeeAlso: AX=0400h,AX=0404h,AX=0407h
-----S-140406-----
INT 14 - Microsoft Systems Journal TSRCOMM INT14 - CLEAR THE TRANSMIT BUFFER
AX = 0406h
Return: nothing
SeeAlso: AX=0400h,AX=0404h,AX=0407h
-----S-140407-----
INT 14 - Microsoft Systems Journal TSRCOMM INT14 - GET TRANSMIT BUFFER COUNT
AX = 0407h
Return: AX = number of characters in the buffer
SeeAlso: AX=0400h,AX=0405h,AX=0406h
-----S-140408-----
INT 14 - Microsoft Systems Journal TSRCOMM INT14 - UNINSTALL
AX = 0408h
Return: nothing
SeeAlso: AX=0400h
-----S-1405-----
INT 14 - SERIAL - EXTENDED COMMUNICATION PORT CONTROL (CONVERTIBLE, PS)
AH = 05h
AL = function
00h read modem control register
Return: BL = modem control register (see #00334)
AH = status
01h write modem control register
BL = modem control register (see #00334)
Return: AX = status
DX = port number
Note: also supported by ArtiCom
SeeAlso: AH=00h, AH=1Fh, AX=8000h"ARTICOM", AH=FBh

Bitfields for modem control register:
Bit(s) Description (Table 00334)
0 data terminal ready
1 request to send
2 OUT1
3 OUT2
4 LOOP
5-7 reserved
-----S-1405-----

INT 14 - FOSSIL - DEINITIALIZE DRIVER

AH = 05h

DX = port number

Return: none

DTR is not affected

SeeAlso: AH=00h,AH=04h"FOSSIL",AH=1Dh,AH=8Dh

-----S-1405-----

INT 14 - MultiDOS Plus IODRV - READ CHARACTER FROM PORT

AH = 05h

AL = timeout in seconds (00h = never)

Return: AL = status

00h successful

AH = character read

01h read error

02h timed out

other modem status (CTS, DSR) changed

Note: the port number is stored at offset BEh in the Task Control Block

SeeAlso: AH=02h,AH=04h"MultiDOS",AH=06h"MultiDOS",AH=22h"MultiDOS"

SeeAlso: INT 15/AH=13h"MultiDOS"

-----S-1405-----

INT 14 - Digiboard DigiCHANNEL PC/X* - CHANGE PROTOCOL

AH = 05h

AL = protocol (see #00335)

BH = new XOFF character (00h = current)

BL = new XON character (00h = current)

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

FFh error

SeeAlso: AH=04h"Digiboard"

Bitfields for Digiboard protocol:

Bit(s) Description (Table 00335)

7-4 unused

3 RTS/CTS

2 DSR

1,0 XON/XOFF

-----S-1405-----

INT 14 - MBBIOS - DROP DTR AND RTS

AH = 05h

DX = port number

Return: none

SeeAlso: AH=00h"MBBIOS",AH=06h"MBBIOS",AH=06h"FOSSIL"

-----S-1405-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - CHANGE PORT PROTOCOL

AH = 05h

AL = new port protocol (see #00336)

BH = new XOFF character

BL = new XON character

DX = port number

Return: AH = FFh if invalid protocol

SeeAlso: AH=00h,AH=04h"SERIAL",AH=06h"PC-MOS"

Bitfields for PC-MOS/386 serial port protocol:

Bit(s) Description (Table 00336)

7 set to enable/disable CD monitoring, clear to set protocol

---bit 7 set---

4 CD monitoring enabled

5 automatic restart enabled

---bit 7 clear---

0 receive XON/XOFF

1 transmit XON/XOFF

2 DTR/DSR

3 RTS/CTS

-----1405-----

INT 14 - PhysTechSoft PTS ROM-DOS - SET PACKET

AH = 05h

CX = packet length

DX = port number

DS:SI -> packet

Return: AH = error code (0 = no error, -1 = error)

Notes: The embedded PC can send messages at any time. The managing PC may force the embedded PC to receive messages only after causing a reset on the remote embedded PC.

!!!from Matthias Paul:

Description of the high-level control packets:

After the reset the embedded PC will send a 4-byte packet "LOAD"

to the managing PC to indicate it is now able to receive commands.

The managing PC can send commands using 6-byte sized packets starting

with "COMM" followed by a 2-byte command code:

"01" demand setup information

"02" demand sending of diagnostic information

If diagnostic information is requested, the embedded PC will send back

a header packet starting with 4-byte "DIAG" followed by 2 bytes indicating the count of diagnostic data packets following. Each diagnostic data packet is sized 134 bytes, starting with the 4-byte "DIAG", a 2-byte number packet, followed by 128 bytes of data.

If setup information is requested, the embedded PC will send back 8 bytes length packets starting with "SETP" and 2 byte of setup information of the LOADER and BIOS. The contents varies.

If the embedded PC requests the managing PC to send a BIOS it will send a 4 byte packet "BIOS".

The managing PC will then send back a 6-byte sized BIOS header packet starting with "BIOS" and a 2-byte indicator of the BIOS image size in packets.

The embedded PC will answer by sending a 6 byte packet starting with "BIOS" followed by the running number of the demanded packet.

The managing PC will then send the requested part of the BIOS image in a 134 byte sized packet, starting with "BIOS", the 2 byte running number of the packet and 128 bytes of the actual BIOS info.

The embedded PC acknowledges that it received the BIOS with a 6-byte packet "BIOS", followed by 2-bytes of 0.

If the embedded PCs requests the managing PC to send the operating system, it will follow the same procedure as for requesting the BIOS. The only difference is that instead of "BIOS", the string "PTOS" will be used in the communication.

Debug information packets have a length of 32 bytes starting with the 4-byte "DEBU" signature, and followed by the contents of the PCU registers in the following order: SP, SS, ES, DS, BP, DI, SI, DX, CX, BX, AX, IP, CS, Flags. Debug information is send after reaching the corresponding breakpoint.

Description of the low-level packet protocol:

Before sending a packet the sending side sends a byte 52h and for a certain time (ca. 1 second for the embedded PC) waits for confirmation (41h) from the receiving side. If no acknowledge is received this procedure is repeated 4 more times before an error is returned.

If acknowledge was received the following information is sent:

2 BYTES synchronization (50h, 50h)

BYTE data packet size - 1

2 BYTES CRC of the data

var. data

2 BYTES synchronization (F0h, F0h)

The receiving side must acknowledge this by sending back two bytes

59h, 59h. If the sender does not receive the confirmation, it
tries 4 more times before returning an error.

SeeAlso: INT 14h/AH=06h, INT 15h/AX=E908h, INT 15h/E909h

-----N-1405--DXFFFF-----

INT 14 - Connection Manager - CLOSE COMMUNICATION

AH = 05h

DX = FFFFh

AL = connection ID

Return: AH = return code

00h successful

01h no such connection

02h invalid connection ID

AL = correct connection ID

Desc: terminate existing connection to allow another one to be established

Note: if DX is 0-3 on entry, Connection Manager emulates the standard BIOS
function, but redirects the port over the network; if DX is any other
value, the call is chained

SeeAlso: AH=04h/DX=FFFFh,AH=0Dh/DX=FFFFh

-----S-1406-----

INT 14 - FOSSIL - RAISE/LOWER DTR

AH = 06h

DX = port

AL = DTR state to be set

00h = lower

01h = raise

Return: nothing

SeeAlso: AH=05h"MBBIOS",AH=1Ah

-----S-1406-----

INT 14 - MultiDOS Plus IODRV - WRITE CHARACTER TO PORT

AH = 06h

AL = character

Return: AL = status

00h successful

Notes: the port number is stored at offset BEh in the Task Control Block
if output queue is full, the calling task is blocked until the
character can be stored

SeeAlso: AH=01h,AH=04h"MultiDOS",AH=05h"MultiDOS",AH=21h"MultiDOS"

SeeAlso: INT 15/AH=13h"MultiDOS"

-----S-1406-----

INT 14 - MBBIOS - RAISE DTR AND RTS

```
AH = 06h
DX = port number
Return: none
SeeAlso: AH=05h"MBBIOS",AH=07h"MBBIOS"
-----S-1406-----
INT 14 - PC-MOS/386 v5.01 $serial.sys v5.04 - DRIVER 'ID' FUNCTION
AH = 06h
DX = port number
Return: AH bit 7 set
AL = number of highest function supported by driver
Program: PC-MOS/386 v5.01 is a multitasking, multiuser MS-DOS 5.0-compatible
operating system by The Software Link, Inc.
SeeAlso: AH=18h"PC-MOS"
-----N-1406-----
INT 14 - TelAPI - WRITE BLOCK
AH = 06h
CX = number of characters to write
DX = port number
ES:DI -> buffer containing data
Return: AX = number of characters actually sent (negative on error)
CX = ???
SeeAlso: AH=07h"TelAPI",AH=E0h"TelAPI",AH=E3h"TelAPI"
-----1406-----
INT 14 - PhysTechSoft PTS ROM-DOS - GET PACKET
AH = 06h
CX = buffer size for packet
DX = port number
DS:SI -> buffer for packet
Return: AH = error code (00h = no error, FFh = error)
CX = packet size
SeeAlso: AH=05h"PTS ROM-DOS",INT 15/AX=E908h,INT 15/AX=E909h
-----N-1406--DXFFFF-----
INT 14 - Connection Manager - SEND CHARACTER BLOCK
AH = 06h
DX = FFFFh
AL = connection ID
CX = number of characters to send
ES:DI -> buffer containing data to be sent
Return: AH = return code (see #00303)
Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network
```

SeeAlso: AH=04h/DX=FFFFh,AH=07h/DX=FFFFh,AH=09h/DX=FFFFh

-----S-1407-----

INT 14 - FOSSIL - RETURN TIMER TICK PARAMETERS

AH = 07h

Return: AL = timer tick interrupt number

AH = ticks per second on interrupt number in AL

DX = approximate number of milliseconds per tick

SeeAlso: AH=16h

-----S-1407-----

INT 14 - MultiDOS Plus IODRV - GET PORT STATUS

AH = 07h

Return: CL = modem status (see #00305)

CH = character at head of input queue (if any)

DX = number of characters in input queue

Note: the port number is stored at offset BEh in the Task Control Block

SeeAlso: AH=03h,AH=05h"MultiDOS",AH=08h"MultiDOS",AH=09h"MultiDOS"

SeeAlso: AH=23h"MultiDOS",INT 15/AH=13h"MultiDOS"

-----S-1407-----

INT 14 - MBBIOS - SEND BREAK

AH = 07h

DX = port number

Return: none

SeeAlso: AH=06h"MBBIOS",AH=FAh"EBIOS"

-----S-1407-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - SEND RS-232 BREAK

AH = 07h

BX = duration of break in clock ticks

DX = port number

Return: nothing

-----N-1407-----

INT 14 - TelAPI - READ BLOCK

AH = 07h

CX = length of buffer in bytes

DX = port number

ES:DI -> buffer for data

Return: AX > 0000h number of characters actually read

AX = 0000h host has closed connection

AX < 0000h error code (see #00397)

CX = ???

Note: translates CRLF into local EOL if the connection is in ASCII mode,
negotiates various Telnet options, and immediately executes several

different Telnet action commands

SeeAlso: AH=06h"TelAPI",AH=E0h"TelAPI",AH=E2h"TelAPI"

-----N-1407--DXFFFF-----

INT 14 - Connection Manager - RECEIVE CHARACTER BLOCK

AH = 07h

DX = FFFFh

AL = connection ID

BL = flag

00h wait for data

nonzero do not wait if no data available

CX = size of receive buffer

ES:DI -> buffer for received characters

Return: AH = return code (00h-02h,04h,FFh) (see #00303)

BH = line status (see #00306)

CX = number of characters received

Program: Connection Manager by Softwarehouse Corp. permits the sharing of serial ports over an IPX or NetBIOS-based network

SeeAlso: AH=01h/DX=FFFFh,AH=04h/DX=FFFFh,AH=06h/DX=FFFFh

-----S-1408-----

INT 14 - FOSSIL - FLUSH OUTPUT BUFFER WAITING TILL ALL OUTPUT IS DONE

AH = 08h

DX = port number

Return: nothing

SeeAlso: AH=09h"FOSSIL"

-----S-1408-----

INT 14 - MultiDOS Plus 4.0 IODRV - GET AND RESET PORT LINE STATUS

AH = 08h

Return: AL = line status (see #00304)

AH destroyed

Notes: the port number is stored at offset BEh in the Task Control Block on every line status change, the line status is ORed with the line status accumulator; this function returns the accumulator and clears it

SeeAlso: AH=03h,AH=04h"MultiDOS",AH=07h"MultiDOS",INT 15/AH=13h"MultiDOS"

-----S-1408-----

INT 14 - Digiboard DigiCHANNEL PC/X* - ALTERNATE STATUS CHECK

AH = 08h

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = RS232 status bits (see #00304 at AH=03h)

ZF set if no characters queued

ZF clear if character available

AL = next character

SeeAlso: AH=03h,AH=08h"PC-MOS",AH=09h"Digiboard",AH=14h"Digiboard"

-----S-1408-----

INT 14 - MBBIOS - NON-DESTRUCTIVE READ

AH = 08h

DX = port number

Return: AL = character (if AH bit 0 set)

AH = status (see #00304)

SeeAlso: AH=0Bh"MBBIOS",AH=0Ch"FOSSIL"

-----S-1408-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - INPUT STATUS CHECK

AH = 08h

DX = port number

Return: CF set if carrier loss detected

ZF set if input buffer empty

ZF clear if characters available

AL = next character dequeued

-----N-1408--DXFFFF-----

INT 14 - Connection Manager - RETURN DEFAULT CONNECTION PARAMETERS

AH = 08h

DX = FFFFh

CX = size of buffer for parameters or 0000h to get length

ES:DI -> buffer for parameter vector (see #00311)

Return: AH = return code

00h successful

CX = number of bytes required (if CX=0000h on entry)

CX = number of bytes omitted for lack of space (if CX nonzero)

nonzero invalid request

Program: Connection Manager by Softwarehouse Corp. permits the sharing of

serial ports over an IPX or NetBIOS-based network

SeeAlso: AH=00h/DX=FFFFh,AH=0Fh/DX=FFFFh

-----S-1409-----

INT 14 - FOSSIL - PURGE OUTPUT BUFFER THROWING AWAY ALL PENDING OUTPUT

AH = 09h

DX = port number

Return: nothing

SeeAlso: AH=08h"FOSSIL",AH=0Ah"FOSSIL",AH=88h

-----S-1409-----

INT 14 - MultiDOS Plus IODRV - RESET PORT STATUS

AH = 09h

Return: modem status byte cleared

Note: the port number is stored at offset BEh in the Task Control Block

SeeAlso: AH=04h"MultiDOS",AH=07h"MultiDOS",INT 15/AH=13h"MultiDOS"

-----S-1409-----

INT 14 - Digiboard DigiCHANNEL PC/X* - CLEAR BUFFERS

AH = 09h

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

FFh error

SeeAlso: AH=08h"Digiboard",AH=0Ah"Digiboard",AH=10h"Digiboard"

-----S-1409-----

INT 14 - MBBIOS - GET/SET OPTIONS

AH = 09h

AL = option byte (see #00337)

DX = port number???

Return: AL = old option byte

SeeAlso: AH=00h"MBBIOS",AH=04h"MBBIOS",AH=10h"FOSSIL"

Bitfields for MBBIOS option byte:

Bit(s) Description (Table 00337)

0 transmit buffering enabled

2 hardware handshaking enabled

5 high-speed option enabled (see AH=00h"MBBIOS",#00302)

other reserved

-----S-1409-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - RESET I/O BUFFER POINTERS

AH = 09h

DX = port number

Return: nothing

SeeAlso: AH=13h"PC-MOS"

-----N-1409--DXFFFF-----

INT 14 - Connection Manager - SEND BREAK

AH = 09h

DX = FFFFh

AL = connection ID

Return: AH = return code (00h-02h) (see #00303 at AH=00h/DX=FFFFh)

Program: Connection Manager by Softwarehouse Corp. permits the sharing of

serial ports over an IPX or NetBIOS-based network

SeeAlso: AH=02h/DX=FFFFh,AH=03h/DX=FFFFh

-----S-140A-----

INT 14 - FOSSIL - PURGE INPUT BUFFER THROWING AWAY ALL PENDING INPUT

```

AH = 0Ah
DX = port number
Return: nothing
SeeAlso: AH=09h"FOSSIL",AH=85h
-----S-140A-----
INT 14 - Digiboard DigiCHANNEL PC/X* - INPUT QUEUE CHECK
AH = 0Ah
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)
Return: AX = number of characters available in buffer
Note: this function is also supported by the PC-MOS/386 v5.01 $serial.sys
SeeAlso: AH=09h"Digiboard",AH=0Dh"Digiboard"
-----S-140A-----
INT 14 - MBBIOS - WRITE BUFFER
AH = 0Ah
CX = count
ES:DI -> buffer (see #00338)
Return: AX = status (see #00304,#00305)
CX = unsend character count
DI updated
Note: the PACCOM version of MBBIOS does not use CX or ES:DI; instead, ES
contains the segment of a buffer containing the packet to be sent,
which by default will be freed once the packet has been sent. Use
AH=0Ch"MBBIOS" to allocate the buffer.
SeeAlso: AH=01h,AH=0Bh"MBBIOS",AH=0Ch"MBBIOS",AH=19h"FOSSIL"

Format of MBBIOS PACCOM buffer:
Offset Size Description (Table 00338)
00h 504 BYTES data area
1F8h WORD length of data in data area
1FAh BYTE flags/status
      bit 7: don't discard buffer after transmitting data
      bit 6: buffer has been transmitted
1FBh BYTE reserved (0) for additional flags/status
1FCh WORD user data
1FEh WORD MBBIOS-internal pointer to next buffer
-----N-140A---DXFFFF-----
INT 14 - Connection Manager - MODIFY ACTIVE CONNECTION PARAMETERS
AH = 0Ah
DX = FFFFh
ES:DI -> vector string containing new parameters (see #00311)
Return: AH = return code (00h-03h,06h) (see #00303)

```

Program: Connection Manager by Softwarehouse Corp. permits the sharing of

serial ports over an IPX or NetBIOS-based network

Note: any subvectors valid for the Change Parameters command replace the

existing values in the current set

SeeAlso: AH=00h/DX=FFFFh,AH=0Fh/DX=FFFFh

-----S-140B-----

INT 14 - FOSSIL - TRANSMIT NO WAIT

AH = 0Bh

AL = character

DX = port number

Return: AX = result

0000h character not accepted

0001h character accepted

SeeAlso: AH=01h

-----S-140B-----

INT 14 - MBBIOS - READ BUFFER

AH = 0Bh

CX = size of buffer

ES:DI -> buffer

Return: AH = composite line status (see #00304) formed by ORing all statuses

on receive interrupts; bit 0 set if additional characters

available

AL = composite modem status (see #00305) formed by ORing all statuses

CX = number of characters actually read

DI updated

Note: the PACCOM version of MBBIOS does not use CX or ES:DI on call,

instead returning ES set to the segment of the buffer containing a

received packet, or 0000h if no packets available; the buffer may

be freed with AH=0Ch"MBBIOS"

SeeAlso: AH=02h,AH=08h"MBBIOS",AH=0Ah"MBBIOS",AH=0Ch"MBBIOS",AH=18h"FOSSIL"

-----N-140B--DXFFFF-----

INT 14 - Connection Manager - PREPARE FOR INBOUND CONNECTION

AH = 0Bh

DX = FFFFh

AL = service name

00h use parameter file or default

01h use specified name

ES:DI -> 16-byte blank-padded name

BH = connection notification

00h program awaiting connection, don't notify user

01h notify user on connecting

BL = connection type
00h connection will use Connection Manager API
Return: AH = return code (00h-02h) (see #00303 at AH=00h/DX=FFFFh)
AL = connection ID if AH=00h
Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network
SeeAlso: AH=04h/DX=FFFFh,AH=0Ch/DX=FFFFh,AH=10h/DX=FFFFh
-----S-140C-----
INT 14 - FOSSIL - NON-DESTRUCTIVE READ AHEAD
AH = 0Ch
DX = port number
Return: AX = FFFFh character not available
AX = 00xxh character xx available
SeeAlso: AH=08h"MBBIOS",AH=20h"FOSSIL"
-----S-140C-----
INT 14 - MBBIOS PACCOM support - BUFFER MANAGEMENT
AH = 0Ch
ES = segment of buffer to free, or 0000h to allocate new buffer
Return: ES = segment of allocated buffer (if ES=0000h on entry)
Note: the PACCOM version of MBBIOS uses only ES as buffer address for
AH=0Ah and AH=0Bh
SeeAlso: AH=0Ah"MBBIOS",AH=0Bh"MBBIOS"
-----N-140C--DXFFFF-----
INT 14 - Connection Manager - TEST FOR INBOUND CONNECTION REQUEST
AH = 0Ch
DX = FFFFh
AL = connection ID from AH=0Bh/DX=FFFFh
Return: AH = return code (00h-03h) (see also #00303 at AH=00h/DX=FFFFh)
03h not prepared for inbound connection
AL = connection ID (if AH=00h) or correct connection ID (if AH=02h)
Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network
SeeAlso: AH=03h/DX=FFFFh,AH=04h/DX=FFFFh,AH=0Bh/DX=FFFFh
-----S-140D-----
INT 14 - FOSSIL - KEYBOARD READ WITHOUT WAIT
AH = 0Dh
Return: AX = result
FFFFh character not available
xxyyh standard IBM-style scan code
SeeAlso: AH=0Eh
-----S-140D-----

INT 14 - Digiboard DigiCHANNEL PC/X* - GET POINTER TO CH_KEY_RDY FLAG

AH = 0Dh

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: ES:BX -> CH_KEY_RDY flag (see #00339)

SeeAlso: AH=0Ah"Digiboard"

(Table 00339)

Values for Digiboard CH_KEY_RDY flag:

00h receive buffer empty

FFh characters available

-----S-140D-----

INT 14 - MBBIOS PACCUM support - SET TXD

AH = 0Dh

AL = new setting (FFh = 1.0)

Return: nothing

Desc: specify the time from RTS to start or packet

SeeAlso: AX=0D00h,AH=0Eh"MBBIOS",AH=0Fh"MBBIOS"

-----N-140D--DXFFFF-----

INT 14 - Connection Manager - TERMINATE CONNECTION CLIENT ACTIVITY

AH = 0Dh

DX = FFFFh

Return: AH = return code

00h successful

nonzero operation not terminated

Program: Connection Manager by Softwarehouse Corp. permits the sharing of serial ports over an IPX or NetBIOS-based network

Desc: end all Connection Client TSR activity to allow it to be removed from memory

SeeAlso: AH=05h/DX=FFFFh,AH=6Fh/BX=FFFFh

-----S-140D00-----

INT 14 - MBBIOS - GET AVAILABLE BYTES

AX = 0D00h

Return: AX = bytes in transmit buffer

CX = bytes in receive buffer

SeeAlso: AH=0Ah"MBBIOS",AH=0Bh"MBBIOS"

-----S-140D01-----

INT 14 - MBBIOS - LOWER ALL MODEM CONTROL SIGNALS

AX = 0D01h

Return: nothing

Note: this function lowers DTR, RTS, etc.

SeeAlso: AX=0D02h

-----S-140D02-----

INT 14 - MBBIOS - RAISE ALL MODEM CONTROL SIGNALS

AX = 0D02h

Return: nothing

Note: this function raises DTR, RTS, etc.

SeeAlso: AX=0D01h

-----S-140D03-----

INT 14 - MBBIOS - SET HANDSHAKE BYTE

AX = 0D03h

CL = new handshake byte

Return: CL = previous handshake byte

Note: this function lowers DTR, RTS, etc.

-----S-140E-----

INT 14 - FOSSIL - KEYBOARD READ WITH WAIT

AH = 0Eh

Return: AX = xxyyh standard IBM-style scan code

SeeAlso: AH=0Dh"FOSSIL"

-----S-140E-----

INT 14 - Digiboard DigiCHANNEL PC/X* - WRITE STRING

AH = 0Eh

CX = number of characters to write

ES:BX -> string

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AX = number of characters actually written

ZF clear if successful

ZF set on error

SeeAlso: AH=0Fh"Digiboard"

-----S-140E-----

INT 14 - MBBIOS PACCUM support - SET PERSISTENCE

AH = 0Eh

AL = new setting (FFh = 1.0)

Return: nothing

Desc: specify the time from end of DCD to RTS

SeeAlso: AH=0Dh"MBBIOS",AH=0Fh"MBBIOS"

-----N-140E--DXFFFF-----

INT 14 - Connection Manager - SET HARDWARE FLOW STATE

AH = 0Eh

DX = FFFFh

AL = connection ID from AH=04h/DX=FFFFh

BL = RTS state (00h off, 01h on)

Return: AH = return code (00h-03h) (see also #00303 at AH=00h/DX=FFFFh)

03h invalid request (BL not 00h or 01h)

Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network

SeeAlso: AH=03h/DX=FFFFh,AH=0Ah/DX=FFFFh

-----S-140F-----

INT 14 - FOSSIL - ENABLE/DISABLE FLOW CONTROL

AH = 0Fh

AL = bit mask describing requested flow control (see #00340)

DX = port number

Return: nothing

SeeAlso: AH=09h"MBBIOS",AH=10h"FOSSIL"

Bitfields for FOSSIL requested flow control:

Bit(s) Description (Table 00340)

0 XON/XOFF on transmit (watch for XOFF while sending)

1 CTS/RTS (CTS on transmit/RTS on receive)

2 reserved

3 XON/XOFF on receive (send XOFF when buffer near full)

4-7 all 1

-----S-140F-----

INT 14 - Digiboard DigiCHANNEL PC/X* - READ STRING

AH = 0Fh

CX = number of characters to read

ES:BX -> buffer

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AX = number of characters read

ZF clear if successful

ZF set on error (line status or wrong number of characters)

SeeAlso: AH=0Eh"Digiboard"

-----S-140F-----

INT 14 - MBBIOS PACCOM support - SET SLOT TIME

AH = 0Fh

AL = new setting in clock ticks

Return: nothing

Desc: specify the time from end of DCD to RTS

SeeAlso: AH=0Dh"MBBIOS",AH=0Eh"MBBIOS",AH=10h"MBBIOS"

-----N-140F--DXFFFF-----

INT 14 - Connection Manager - RETURN ACTIVE CONNECTION PARAMETERS

AH = 0Fh

DX = FFFFh

AL = connection ID

CX = size of buffer or 0000h to get length of returned vector
ES:DI -> buffer for connection parameter vector (see #00311)
Return: AH = return code (00h-02h,06h) (see #00303 at AH=00h/DX=FFFFh)
CX = number of bytes which could not be returned because the given
buffer was too small
Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network
SeeAlso: AH=08h/DX=FFFFh,AH=0Ah/DX=FFFFh
-----S-1410-----
INT 14 - FOSSIL - EXTENDED ^C/^K CHECKING AND TRANSMIT ON/OFF
AH = 10h
AL = bit mask
bit 0: enable/disable ^C/^K checking
bit 1: enable/disable the transmitter
DX = port number
Return: nothing
SeeAlso: AH=0Fh"FOSSIL"
-----S-1410-----
INT 14 - Digiboard DigiCHANNEL PC/X* - CLEAR RECEIVE BUFFER
AH = 10h
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)
Return: AH = status
00h successful
FFh error
SeeAlso: AH=09h"Digiboard",AH=11h"Digiboard"
-----S-1410-----
INT 14 - MBBIOS PACC0M support - SET CRC WAIT
AH = 10h
AL = new setting in clock ticks (should be at least 5 character times)
Return: nothing
Desc: specify the time from start of last character to dropping RTS
SeeAlso: AH=0Dh"MBBIOS",AH=0Fh"MBBIOS"
-----N-1410--DXFFFF-----
INT 14 - Connection Manager - QUERY SERVICE NAMES
AH = 10h
DX = FFFFh
CL = subfunction
00h search first
01h search next
ES:DI -> pattern buffer (see #00341)
Return: AH = return code (00h,01h,03h,06h) (see also #00303 at AH=00h/DX=FFFFh)

01h no (more) matching names

03h invalid request

ES:DI buffer filled with reply buffer (see #00341) containing matched
name if AH=00h

Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network

Desc: obtain the names of groups and lines available for connection requests,
and the names of active Connection Servers

SeeAlso: AH=04h/DX=FFFFh,AH=0Bh/DX=FFFFh

Format of Connection Manager pattern/reply buffer:

Offset Size Description (Table 00341)

00h WORD length of pattern (30h or 32h)

02h 16 BYTES server pattern or name

12h 16 BYTES group pattern or name

22h 16 BYTES line pattern or name

23h BYTE (optional) ???

24h BYTE (optional, returned) current line status

00h available

01h out of service

02h currently allocated to a connection

Note: pattern may include '?' wildcard to match any character

-----S-1411-----

INT 14 - FOSSIL - SET CURRENT CURSOR LOCATION

AH = 11h

DH = row

DL = column

Return: nothing

Note: this is the same as INT 10/AH=02h

SeeAlso: AH=12h"FOSSIL"

-----S-1411-----

INT 14 - Digiboard DigiCHANNEL PC/X* - CLEAR TRANSMIT BUFFER

AH = 11h

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

FFh error

SeeAlso: AH=09h"Digiboard",AH=10h"Digiboard"

-----S-1411-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - DISABLE PORT

AH = 11h

```
DX = port number
Return: AL = status
    00h successful
    01h IRQ for port is shared
    02h IRQ was reserved
SeeAlso: AH=04h"SERIAL",AH=05h"SERIAL",AH=12h"PC-MOS"
-----S-1412-----
INT 14 - FOSSIL - READ CURRENT CURSOR LOCATION
    AH = 12h
Return: DH = row
    DL = column
Note: this is the same as INT 10/AH=03h
SeeAlso: AH=11h"FOSSIL"
-----S-1412-----
INT 14 - Digiboard DigiCHANNEL PC/X* - GET TRANSMIT BUFFER FREE SPACE
    AH = 12h
    DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)
Return: AX = number of bytes free
SeeAlso: AH=0Ah"Digiboard",AH=14h"Digiboard"
-----S-1412-----
INT 14 - PC-MOS/386 v5.01 $serial.sys v5.04 - GET CURRENT PORT PARAMETERS
    AH = 12h
    DX = port number
Return: AH = status
    FFh port number invalid
    AL = line parameters (see #00304)
    AH = flow control configuration (see #00336 at AH=05h"PC-MOS")
    CX:BX = bps rate
    DL = XOFF character or 00h for none
    DH = XON character or 00h for none
-----S-1413-----
INT 14 - FOSSIL - SINGLE CHARACTER ANSI WRITE TO SCREEN
    AH = 13h
    AL = character
Return: nothing
Note: should not be called if it is unsafe to call DOS
SeeAlso: AH=15h
-----S-1413-----
INT 14 - PC-MOS/386 v5.01 $serial.sys v5.04 - REGISTER A PORT WITH A TERMINAL
    AH = 13h
    DX = port number
```

Return: AH = status
 FFh port number invalid
 else
 ES:BX -> BYTE flag (00h buffer empty, FFh buffer contains data)

SeeAlso: AH=17h"PC-MOS"

-----S-1414-----

INT 14 - FOSSIL - ENABLE OR DISABLE WATCHDOG PROCESSING

AH = 14h
AL = 01h enable watchdog
 00h disable watchdog
DX = port number

Return: nothing

SeeAlso: INT 21/AH=2Bh/CX=6269h"WDTSR"

-----S-1414-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - OUTPUT STRING

AH = 14h
CX = number of characters in string
DX = port number
ES:BX -> string to be sent
SI = timeout in timer ticks or 0000h for default

Return: AX = number of bytes actually sent

ZF clear if successful
ZF set on timeout

SeeAlso: AH=01h,AH=15h"PC-MOS"

-----S-1414-----

INT 14 - Digiboard - GET NUMBER OF BOARDS INSTALLED

AH = 14h
Return: AX = number of boards installed

SeeAlso: AH=08h"Digiboard",AH=15h"Digiboard"

-----S-1415-----

INT 14 - FOSSIL - WRITE CHARACTER TO SCREEN USING BIOS SUPPORT ROUTINES

AH = 15h
AL = character

Return: nothing

SeeAlso: AH=13h"FOSSIL"

-----S-1415-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - INPUT STRING

AH = 15h
CX = size of buffer
DX = port number
ES:BX -> buffer for received characters

SI = timeout in clock ticks or 0000h for default
Return: AX = number of characters actually read
ZF set on timeout (no data available)
SeeAlso: AH=02h,AH=14h"PC-MOS",AH=16h"PC-MOS"
-----S-1415-----
INT 14 - Digiboard - ENABLE/DISABLE MEMORY
AH = 15h
AL = new state (00h disabled, 01h enabled)
Return: AH = status
00h successful
80h error
FFh error
SeeAlso: AH=14h"Digiboard",AH=16h"Digiboard"
-----S-1416-----
INT 14 - FOSSIL - INSERT/DELETE FUNCTION FROM TIMER TICK CHAIN
AH = 16h
AL = function
00h = delete
01h = add
ES:DX -> routine to call
Return: AX = status
0000h successful
0001h unsuccessful
SeeAlso: AH=07h"FOSSIL"
-----S-1416-----
INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - LINK TO ANOTHER SERIAL DRIVER
AH = 16h
ES:BX -> calling driver's INT 14 entry point
Return: nothing
Program: PC-MOS/386 v5.01 is a multitasking, multiuser MS-DOS 5.0-compatible
operating system by The Software Link, Inc.
-----S-1416-----
INT 14 - Digiboard DigiCHANNEL PC/X* - CCB COMMAND
AH = 16h
AL = CCB command number (see #00342) (see also following entries)
BL = byte 2
BH = byte 3
CL = byte 1 (for all channel functions except 4Eh and 4Fh)
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)
Return: AH = status
00h successful

80h error

FFh error

SeeAlso: AX=1646h,AH=18h"Digiboard"

(Table 00342)

Values for Digiboard CCB command number:

40h Set Receive Mid Water Mark
 41h Set Receive High Water Mark
 42h Flush Receive Buffer
 43h Flush Transmit Buffer
 44h Transmit Pause
 45h Transmit Resume
 46h Set Interrupt to Host Mask
 47h Set Baud, Data, Stop and Parity
 48h Send Break
 49h Set Modem Lines
 4Ah Set Break Count
 4Bh Set Handshake
 4Ch Set Xon/Xoff Characters
 4Dh Set Transmit Mid Water Mark
 4Eh IRQ Polling Timer to Host
 4Fh Buffer Set All
 50h Port On
 51h Port Off
 52h Receive Pause
 53h Special Character Interrupt
 54h RS-422 Enable

-----S-141646-----

INT 14 - Digiboard - CCB COMMAND - SET INTERRUPT TO HOST MASK

AX = 1646h
 BL = bits to set
 BH = bits to clear
 CL = byte 1
 DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful
 80h error
 FFh error

SeeAlso: AH=16h"Digiboard",AX=1647h

-----S-141647-----

INT 14 - Digiboard - CCB COMMAND - SET BAUD/DATABITS/STOPBITS/PARITY

AX = 1647h
BL = baud
BH = datatype
CL = byte 1
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status
00h successful
80h error
FFh error

SeeAlso: AH=16h"Digiboard",AX=1646h,AX=1649h

-----S-141649-----

INT 14 - Digiboard - CCB COMMAND - SET MODEM LINES

AX = 1649h
BL = bits to set
BH = bits to clear
CL = byte 1
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status
00h successful
80h error
FFh error

SeeAlso: AH=16h"Digiboard",AX=1647h

-----S-14164A-----

INT 14 - Digiboard - CCB COMMAND - SET BREAK COUNT

AX = 164Ah
BL = break count
CL = byte 1
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status
00h successful
80h error
FFh error

SeeAlso: AH=16h"Digiboard",AX=1649h,AX=164Bh

-----S-14164B-----

INT 14 - Digiboard - CCB COMMAND - SET HANDSHAKE

AX = 164Bh
BL = bits to set
BH = bits to clear
CL = byte 1
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

80h error

FFh error

SeeAlso: AH=16h"Digiboard",AX=1649h,AX=164Ch

-----S-14164C-----

INT 14 - Digiboard - CCB COMMAND - SET XON/XOFF CHARACTERS

AX = 164Ch

BL = XON character

BH = XOFF character

CL = byte 1

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

80h error

FFh error

SeeAlso: AH=16h"Digiboard",AX=164Bh,AX=164Dh

-----S-14164D-----

INT 14 - Digiboard - CCB COMMAND - SET TRANSMIT MID-WATER MARK

AX = 164Dh

BX = new mid-water mark

CL = byte 1

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

80h error

FFh error

SeeAlso: AH=16h"Digiboard",AX=164Ch,AX=164Eh,AX=164Fh

-----S-14164E-----

INT 14 - Digiboard - CCB COMMAND - IRQ POLLING TIMER TO HOST

AX = 164Eh

BL = ticks

BH = ???

CL = mode

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status

00h successful

80h error

FFh error

SeeAlso: AH=16h"Digiboard",AX=164Dh

-----S-14164F-----

INT 14 - Digiboard - CCB COMMAND - BUFFER SET ALL

AX = 164Fh
BL = size
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status
00h successful
80h error
FFh error

SeeAlso: AH=16h"Digiboard",AX=164Dh

-----S-141653-----

INT 14 - Digiboard - CCB COMMAND - SPECIAL CHARACTER INTERRUPT

AX = 1653h
BL = enable/disable
BH = special character
CL = byte 1
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: AH = status
00h successful
80h error
FFh error

SeeAlso: AH=16h"Digiboard",AX=1646h

-----S-1417-----

INT 14 - FOSSIL - REBOOT SYSTEM

AH = 17h
AL = method
00h = cold boot
01h = warm boot

SeeAlso: INT 16/AX=E0FFh,INT 19,INT 60/DI=0606h

-----S-1417-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - WRITE MODEM CONTROL REGISTER

AH = 17h
AL = new value for UART's modem control register
DX = port number

Return: nothing

-----S-1418-----

INT 14 - FOSSIL - READ BLOCK

AH = 18h
CX = maximum number of characters to transfer
DX = port number
ES:DI -> user buffer

Return: AX = number of characters transferred

SeeAlso: AH=19h"FOSSIL",AH=83h"COURIERS",AX=FF02h,INT 6B/AX=0100h

-----S-1418-----
INT 14 - Digiboard DigiCHANNEL PC/X* - SEND BIOS COMMAND
AH = 18h
ES:BX -> 16-byte command string
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)
Return: AH = status
00h successful
80h timeout
AL = mailbox status
00h no errors
8Xh BIOS error
ES:BX buffer filled in with mailbox string
ZF clear if no errors
ZF set if either status byte contains an error code
SeeAlso: AH=16h"Digiboard"

-----S-1418-----
INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - GET DRIVER DESCRIPTION
AH = 18h
DX = port number
Return: DS:BX -> 40-byte buffer containing a string identifying the serial
driver
SeeAlso: AH=06h"PC-MOS"

-----S-1419-----
INT 14 - FOSSIL - WRITE BLOCK
AH = 19h
CX = maximum number of characters to transfer
DX = port number
ES:DI -> user buffer
Return: AX = number of characters transferred
SeeAlso: AH=18h"FOSSIL",AH=86h,INT 6B/AX=0000h

-----S-1419-----
INT 14 - Digiboard DigiCHANNEL PC/X* - SPECIAL CHARACTER INTERRUPT
AH = 19h
BL = flag
00h disable special character interrupt
FFh enable interrupt
DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)
Return: AH = status
00h successful
FFh failed
SeeAlso: AH=1Ah"Digiboard"

-----S-1419-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - SELECTIVE BUFFER FLUSH

AH = 19h

AL = what to flush

bit 0: input buffer

bit 1: output buffer

DX = port number

Return: nothing

SeeAlso: AH=09h"PC-MOS"

-----S-141A-----

INT 14 - FOSSIL - BREAK BEGIN OR END

AH = 1Ah

AL = 00h stop sending 'break'

01h start sending 'break'

DX = port number

Return: nothing

SeeAlso: AH=06h"FOSSIL",AH=8Ah,AH=FAh

-----S-141A-----

INT 14 - Digiboard DigiCHANNEL PC/X - SPECIAL CHARACTER FLAG/COUNTER

AH = 1Ah

BX = subfunction

00h return pointer to special character flag byte

01h return pointer to special character counter word

DX = port number (00h-03h) (04h-43h for XAPCM232.SYS)

Return: ES:BX -> special character flag or counter

Notes: flag is FFh if one or more special characters are in the receive
buffer; it is 00h and the counter is invalid if no special characters
are in the receive buffercounter (if valid) contains the number of characters in the receive
buffer up to and including the last-received special character

-----S-141B-----

INT 14 - FOSSIL - RETURN INFORMATION ABOUT THE DRIVER

AH = 1Bh

DX = port number

CX = size of user buffer

ES:DI -> user buffer for driver info (see #00343)

Return: AX = number of characters transferred

CX = 3058h ("0X") (X00 FOSSIL only)

DX = 2030h (" 0") (X00 FOSSIL only)

Format of FOSSIL driver info:

Offset Size Description (Table 00343)

00h WORD size of structure in bytes
02h BYTE FOSSIL spec driver conforms to
03h BYTE revision level of this specific driver
04h DWORD pointer to ASCIZ identification string
08h WORD size of the input buffer
0Ah WORD number of bytes left in buffer
0Ch WORD size of the output buffer
0Eh WORD number of bytes left in buffer
10h BYTE width of screen
11h BYTE length of screen
12h BYTE actual baud rate, computer to modem

-----S-141C-----

INT 14 - X00 FOSSIL - ACTIVATE PORT

AH = 1Ch

DX = port number

Return: AX = 1954h if successful

BL = maximum function number supported (not including 7Eh and above)

BH = revision of FOSSIL specification supported

Note: this is a duplicate of AH=04h, so that AH=04h may be made compatible
with the PS/2 BIOS in a future release

SeeAlso: AH=04h"FOSSIL",AH=1Dh

-----S-141D-----

INT 14 - X00 FOSSIL - DEACTIVATE PORT

AH = 1Dh

DX = port number

Return: none

Notes: this is a duplicate of AH=05h, so that AH=05h may be made compatible
with the PS/2 BIOS in a future release

ignored if the port was never activated with AH=04h or AH=1Ch

SeeAlso: AH=05h"FOSSIL",AH=1Ch

-----S-141E-----

INT 14 - X00 FOSSIL - EXTENDED LINE CONTROL INITIALIZATION

AH = 1Eh

AL = break status

00h if break

01h if no break

BH = parity (see #00344)

BL = number of stop bits

00h one stop bit

01h two stop bits (1.5 if 5 bit word length)

CH = word length (see #00345)

CL = bps rate (see #00346)

DX = port number

Return: AX = port status code (see #00304,#00305)

Notes: this function is intended to exactly emulate the PS/2 BIOS AH=04h call
if the port was locked at X00 load time, the appropriate parameters are
ignored

SeeAlso: AH=00h,AH=04h"SERIAL"

(Table 00344)

Values for X00 FOSSIL parity:

00h no parity

01h odd parity

02h even parity

03h stick parity odd

04h stick parity even

SeeAlso: #00307,#00345,#00346

(Table 00345)

Values for X00 FOSSIL word length:

00h 5 bits

01h 6 bits

02h 7 bits

03h 8 bits

SeeAlso: #00308,#00344,#00346

(Table 00346)

Values for X00 FOSSIL bps rate:

00h 110

01h 150

02h 300

03h 600

04h 1200

05h 2400

06h 4800

07h 9600

08h 19200

SeeAlso: #00309,#00344,#00345

-----S-141E-----

INT 14 - HUNTER 16 - READ COMMS PARAMETERS

AH = 1Eh

CX = channel number (00h COM1, 01h COM2, ...)

DS:BX -> buffer for communications parameters (see #00347)

Return: DS:BX buffer filled

Note: the Husky Hunter 16 is an 8088-based ruggedized laptop. Other family members are the Husky Hunter, Husky Hunter 16/80, and Husky Hawk.

SeeAlso: AH=20h"HUNTER"

Format of HUNTER 16 communications parameters:

Offset Size Description (Table 00347)

00h	BYTE	communications type (00h IBM, nonzero Husky)
01h	BYTE	port number (00h COM1, 01h COM2)
02h	BYTE	baud rate (00h 110 bps, 01h 150, 02h 300, 03h 600, 04h 1200, 05h 2400, 06h 4800, 07h 9600, 08h 19200, 09h 38400)
03h	BYTE	data bits (01h seven, 02h eight)
04h	BYTE	parity (00h none, 01h odd, 02h even)
05h	BYTE	stop bits (00h one, 01h two)
06h	BYTE	handshake (see #00348)
07h	BYTE	handshake protocol (00h none, 01h Xon/Xoff, 02h HWK3780)
08h	BYTE	Nulls after CR (0-20)
09h	BYTE	LF (00h off, 01h on)
0Ah	BYTE	Serig. 0..7Fh: Ignore this character 80h: Serig off
0Bh	BYTE	echo (00h off, 01h on)
0Ch	BYTE	transmit timeout in seconds (1-60) or 00h to disable
0Dh	BYTE	receive timeout in seconds (1-60) or 00h to disable
0Eh	5 BYTES	reserved

Bitfields for HUNTER 16 handshake:

Bit(s) Description (Table 00348)

0-1	0: RTS off, 1: RTS hold, 2: RTS true
2	DTR enabled
4	CTS enabled
5	DSR enabled
6	DCD enabled

-----S-141F-----

INT 14 - X00 FOSSIL - EXTENDED SERIAL PORT STATUS/CONTROL

AH = 1Fh

DX = port number

AL = direction

00h read modem control register

Return: BL = modem control register (see #00349)

AH = status
01h write modem control register
BL = modem control register (see #00349)
Return: AX = status

Notes: this function is intended to exactly emulate the PS/2 BIOS AH=05h call
X00 forces BL bit 3 set (interrupts cannot be disabled)
SeeAlso: AH=00h,AH=05h"SERIAL"

Bitfields for X00 FOSSIL modem control register:

Bit(s) Description (Table 00349)

0 data terminal ready
1 request to send
2 OUT1
3 OUT2 (interrupts) enabled
4 LOOP
5-7 reserved

-----S-1420-----

INT 14 - X00 FOSSIL - DESTRUCTIVE READ WITH NO WAIT

AH = 20h
DX = port number

Return: AH = 00h if character was available

AL = next character (removed from receive buffer)

AX = FFFFh if no character available

SeeAlso: AH=0Ch,AH=21h"X00"

-----S-1420-----

INT 14 - Alloy MW386 - ATTACH LOGICAL COMMUNICATIONS PORT TO PHYSICAL PORT

AH = 20h
AL = logical port (01h COM1, 02h COM2)
DX = physical port number

Return: AX = status

0000h successful

FFFFh failed

SeeAlso: AH=21h"Alloy",AH=22h"Alloy",AH=23h"Alloy",INT 17/AH=8Bh"Alloy"

-----S-1420-----

INT 14 - MultiDOS Plus - INITIALIZE PORT

AH = 20h
AL = port parameters (see #00300 at AH=00h"SERIAL")
DX = port number (0-3)

Return: AH = status

00h successful

41h no such port

64h monitor mode already active

SeeAlso: AH=00h"SERIAL",AH=04h"MultiDOS",AH=21h"MultiDOS",AH=23h"MultiDOS"

-----S-1420-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - CHECK OUTPUT QUEUE

AH = 20h

DX = port number

Return: AX = number of bytes in output buffer

SeeAlso: AH=0Ah"Digiboard"

-----S-1420-----

INT 14 - HUNTER 16 - SELECT COMMS PARAMETERS

AH = 20h

CX = channel number (00h COM1, 01h COM2, ...)

DS:BX -> buffer with communications parameter (see #00347)

Return: AX = status

0000h successful

0001h invalid parameter

SeeAlso: AH=1Eh"HUNTER",AH=21h"HUNTER"

-----S-1421-----

INT 14 - X00 FOSSIL - STUFF RECEIVE BUFFER

AH = 21h

AL = character

DX = port number

Return: nothing

Notes: the given character is inserted at the end of the receive buffer as if

it had just arrived from the serial port; all normal receive

processing (XON/XOFF, ^C/^K) is performed on the character

fully re-entrant

SeeAlso: AH=20h"X00"

-----S-1421-----

INT 14 - Alloy MW386 v1.x only - RELEASE PHYSICAL COMMUNICATIONS PORT

AH = 21h

DX = physical port number

Return: AX = status

0000h successful

FFFFh failed

SeeAlso: AH=20h"Alloy",AH=22h"Alloy"

-----S-1421-----

INT 14 - MultiDOS Plus - TRANSMIT CHARACTER

AH = 21h

AL = character to send

DX = port number

Return: AH = status (see #00350)

Note: monitor mode must have been turned on with AH=24h before calling

SeeAlso: AH=20h"MultiDOS",AH=22h"MultiDOS",AH=24h"MultiDOS"

(Table 00350)

Values for MultiDOS Plus status:

00h successful
39h no DSR or CTS
3Ch no DSR
3Bh no CTS
41h no such port
42h monitor mode not active
97h timed out

-----S-1421-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - OUTPUT CHARACTER, WITH TIMEOUT

AH = 21h
AL = char to send
DX = port number
SI = timeout in timer ticks (0000h = default)

Return: after character is sent or timeout expires

Program: PC-MOS/386 v5.01 is a multitasking, multiuser MS-DOS 5.0-compatible
operating system by The Software Link, Inc.

SeeAlso: AH=01h,AH=0Eh"Digiboard",AH=22h"PC-MOS"

-----S-1421-----

INT 14 - HUNTER 16 - EXTENDED CONTROL

AH = 21h
AL = command
01h force transmission of buffer
02h clear transmit buffer
03h clear receive buffer
DX = port (00h COM1, 01h COM2)

Return: AH = extended status

Desc: executes the command on the selected port

SeeAlso: AH=1Eh"HUNTER",AH=20h"HUNTER",AH=22h"HUNTER",AH=24h"HUNTER"

-----S-1422-----

INT 14 - Alloy MW386 v2+ - RELEASE LOGICAL COMMUNICATIONS PORT

AH = 22h
AL = logical port (01h COM1, 02h COM2)

Return: AX = status (0000h successful)

SeeAlso: AH=20h"Alloy",AH=21h"Alloy"

-----S-1422-----

INT 14 - MultiDOS Plus - RECEIVE CHARACTER

AH = 22h

DX = port number

Return: AH = status (see also AH=21h"MultiDOS")

00h successful

AL = character

3Dh framing and parity error

3Eh overrun error

3Fh framing error

40h parity error

96h ring buffer overflow

Note: if no character is available, this function waits until a character arrives or an implementation-dependent timeout elapses

SeeAlso: AH=20h"MultiDOS",AH=21h"MultiDOS",AH=27h

-----S-1422-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - RECEIVE CHARACTER, WITH TIMEOUT

AH = 22h

DX = port number

SI = timeout in timer ticks (0000h = default)

Return: AH = port status (see also #00304 at AH=03h)

bit 7 = 1 indicates time-out

AL = character received

Program: PC-MOS/386 v5.01 is a multitasking, multiuser MS-DOS 5.0-compatible operating system by The Software Link, Inc.

SeeAlso: AH=02h,AH=0Fh"Digiboard",AH=21h"PC-MOS"

-----S-1422-----

INT 14 - HUNTER 16 - EXTENDED STATUS

AH = 22h

DX = port (00h COM1, 01h COM2)

Return: AH = extended status

BX = number of characters in input buffer

CX = number of characters in output buffer

Desc: returns the most recent Extended Status code for the port

SeeAlso: AH=21h"HUNTER"

-----S-1423-----

INT 14 - Alloy MW386 v2+ - GET PORT NUMBER FROM LOGICAL PORT ID

AH = 23h

AL = logical port (01h COM1, 02h COM2)

DH = user ID

DL = process ID (DH,DL both FFh for current task)

Return: AL = MW386 port mode (see #00351)

CX = MW386 port number

DH = owner's user ID

DL = owner's task ID

SeeAlso: AH=20h"Alloy",INT 17/AH=8Bh"Alloy"

Bitfields for MW386 port mode:

Bit(s) Description (Table 00351)

0 port is shared (spooler only)

1 port is spooled instead of direct (spooler only)

2 port is assigned as logical COM device, not in spooler

3 port is free

-----S-1423-----

INT 14 - MultiDOS Plus - GET PORT STATUS

AH = 23h

DX = port number

Return: AH = line status (see #00304 at AH=03h)

AL = modem status (see #00305 at AH=03h)

SeeAlso: AH=03h,AH=07h"MultiDOS",AH=20h"MultiDOS"

-----S-1423-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - DECLARE PORT OWNERSHIP

AH = 23h

DX = port number

BX = TCB segment/selector address of owner task

Return: nothing

SeeAlso: AH=0Dh"Digiboard",AH=21h"PC-MOS",AH=22h"PC-MOS"

-----S-1423-----

INT 14 - HUNTER 16 - CONTROL HANDSHAKE LINES

AH = 23h

BH = handshake line to set (00h RTS, 01h DTR)

BL = new level (00h low, 01h high)

Return: nothing

Desc: sets the handshake lines of COM1 to the desired level

SeeAlso: AH=21h"HUNTER",AH=24h"HUNTER"

-----S-1424-----

INT 14 - Alloy MW386 v2+ - CHANGE PHYSICAL PORT PARAMETERS

AH = 24h

CX = physical I/O port number

DS:DX -> configuration table (see #00352)

Return: AH = 00h

Note: invalid port numbers are merely ignored

SeeAlso: INT 17/AH=96h

Format of Alloy MW386 configuration table:

Offset Size Description (Table 00352)

00h	BYTE	baud rate (see #00353)
01h	BYTE	data bits (00h=5, 01h=6, 02h=7, 03h=8)
02h	BYTE	parity (00h none, 01h odd, 02h even)
03h	BYTE	stop bits (00h=1, 01h=2)
04h	BYTE	receive flow control
		00h none, 01h XON/XOFF, 02h DTR/DSR, 03h XPC, 04h RTS/CTS
05h	BYTE	transmit flow control (as for receive)

(Table 00353)

Values for Alloy MW386 baud rate:

00h	38400
01h	19200
02h	9600
03h	7200
04h	4800
05h	3600
06h	2400
07h	2000
08h	1200
09h	600
0Ah	300
0Bh	150
0Ch	134.5

SeeAlso: #00309

-----S-1424-----

INT 14 - MultiDOS Plus - SET MONITOR MODE

AH = 24h

AL = port status storage

00h single status for entire receive buffer

01h separate status kept for each byte in receive buffer

DX = port number

Return: AH = status

00h successful

3Ah invalid status storage specified

41h no such port

64h monitor mode already active

Note: in monitor mode, MultiDOS redirects all BIOS video output to a serial port

SeeAlso: AH=20h"MultiDOS",AH=25h

-----S-1424-----

INT 14 - PC-MOS/386 v5.01 \$serial.sys v5.04 - ???

AH = 24h

Return: ???

Program: PC-MOS/386 v5.01 is a multitasking, multiuser MS-DOS 5.0-compatible
operating system by The Software Link, Inc.

-----S-1424-----

INT 14 - HUNTER 16 - CONTROL CTS HANDSHAKING

AH = 24h

AL = new CTS handshake state for COM1 (00h disabled, 01h enabled)

Return: nothing

SeeAlso: AH=23h"HUNTER",AH=25h"HUNTER",AH=26h"HUNTER"

-----S-1425-----

INT 14 - MultiDOS Plus - CLEAR BUFFERS

AH = 25h

AL = function

00h only clear buffers

01h clear buffers and deactivate

DX = port number

Return: AH = status

00h successful

3Ah invalid function

41h no such port

42h monitor mode not active

SeeAlso: AH=20h"MultiDOS",AH=24h"MultiDOS"

-----S-1425-----

INT 14 - HUNTER 16 - CONTROL RS232 DRIVERS

AH = 25h

AL = new state of RS232 drivers (00h off, 01h on)

Return: nothing

Note: this function can be used to save power by turning off the RS232
drivers. It can also be used to turn on the RS232 drivers before
connecting to a remote system to avoid "garbage" while the drivers
turn on.

SeeAlso: AH=23h"HUNTER",AH=24h"HUNTER",AH=26h"HUNTER"

-----S-1426-----

INT 14 - HUNTER 16 - CONTROL RI POWER UP

AH = 26h

AL = 00h enable RI power up

else disable RI power up

```
Return: nothing
Desc: control whether the Ring Indicator handshake can power up the Hunter
-----S-1427-----
INT 14 - MultiDOS Plus - GET BUFFER CHARACTER COUNT
  AH = 27h
  DX = port number
Return: AH = status
  00h successful
  41h no such port
  42h monitor mode not active
  AL = number of characters in receive buffer
-----S-1427-----
INT 14 - HUNTER 16 - GET INSTALLED PROTOCOLS COUNT
  AH = 27h
  AL = number of extended protocols installed (since last call)
Return: AL = total number installed, including new ones
Desc: Returns the number of extended communication protocols installed
SeeAlso: AH=25h"HUNTER",AH=28h"HUNTER"
-----S-1428-----
INT 14 - HUNTER 16 - GET PROTOCOL NAME
  AH = 28h
  AL = protocol handle
  DS:BX -> 8 character buffer for protocol name
Return: AH = status
  00h successful
  DS:BX buffer filled with the protocol name
  FFh failed
SeeAlso: AH=27h"HUNTER",AH=29h"HUNTER"
-----S-1429-----
INT 14 - HUNTER 16 - GET PROTOCOL HANDLE
  AH = 29h
  DS:BX -> buffer containing the protocol name
Return: AH = status
  00h successful
  AL = handle
  FFh failed
SeeAlso: AH=28h"HUNTER",AH=2Ah"HUNTER"
-----S-142A-----
INT 14 - HUNTER 16 - EXTENDED PROTOCOL MENU
  AH = 2Ah
  AL = protocol handle
```

```
Return: AH = status
        00h successful
        FFh failed
AL = menu handle
SeeAlso: AH=29h"HUNTER",AH=2Bh"HUNTER"
-----S-142B-----
INT 14 - HUNTER 16 - GET EXTENDED PROTOCOL PARAMETERS
        AH = 2Bh
        AL = protocol handle
        DS:BX -> buffer for extended protocol parameters
Return: AH = status
        00h successful
        DS:BX buffer filled with extended parameters
        FFh failed
SeeAlso: AH=2Ah"HUNTER"
-----S-142C00-----
INT 14 - HUNTER 16 - GET DTR
        AX = 2C00h
Return: AH = 00h
        BL = current state
            00h normal DTR operation
            else DTR is forced high
Desc: Indicates whether the DTR signal on COM1 is forced high
Note: the Husky Hunter 16 is an 8088-based ruggedized laptop. Other family
      members are the Husky Hunter, Husky Hunter 16/80, and Husky Hawk.
SeeAlso: AH=2Ch"SET DTR"
-----S-142C-----
INT 14 - HUNTER 16 - SET DTR
        AH = 2Ch
        AL nonzero
        BL = new state
            00h normal operation
            else force DTR high
Return: AH = 00h
Desc: determine whether the DTR signal on COM1 should be forced high
SeeAlso: AX=2C00h
-----N-1436-----
INT 14 - ComShare - INSTALLATION CHECK
        AH = 36h
Return: AX = 4353h ('CS') if installed
        BX = bitmap of installed ports (bit 0: COM1 is gateway, etc.)
```

```
CX reserved for future use
WORD DX:[0100h] = ComShare version number
Program: The ComShare System is a modem-sharing program for NetBIOS and
NetWare-based networks by NashaKala Corporation
Note: ComShare supports the standard BIOS INT 14h calls with a slight
change in the interpretation of speed values
(see #00309 at AH=04h"SERIAL")
SeeAlso: AH=00h"SERIAL",AH=04h"SERIAL",AX=F4FFh
-----t-144000-----
INT 14 - I1541 - INSTALLATION CHECK
AX = 4000h
Return: AX = 1541h if installed
BH = I1541 major version (BCD)
BL = I1541 minor version (BCD)
CX = 0000h
-----t-144001-----
INT 14 - I1541 - TEST IF 1541 CABLE CONNECTED
AX = 4001h
Return: CF clear if connected
BL = LPT number where 1541 cable is connected (1..3)
CF set if cable not connected
Desc: scan all the LPT ports searching for the adapter cable
SeeAlso: AX=4000h
-----t-144002-----
INT 14 - I1541 - SELECT LPT PORT FOR OUTPUT
AX = 4002h
BL = LPT number (1..3)
Return: CF clear if successful
CF set otherwise
Desc: force the input output routines to work on the cable placed on LPT BL
SeeAlso: AX=4001h
-----t-144003-----
INT 14 - I1541 - RESET ALL DEVICES
AX = 4003h
Return: nothing
Desc: send a reset pulse of 100ms to all CBM devices
Note: it is necessary to wait about 2 seconds after reset before executing
other instructions
SeeAlso: AX=4000h,AX=4004h
-----t-144004-----
INT 14 - I1541 - SEND LISTEN SIGNAL
```

AX = 4004h
BH = device number (0..15)
Return: CF clear if successful
CF set on error
AL = error number (see #00354)
SeeAlso: AX=4005h,AX=4006h

(Table 00354)

Values for I1541 error number:

00h device not present
01h listener not ready
02h missing EOI time-out
03h EOI not completed
04h data not released
05h frame error

-----t-144005-----
INT 14 - I1541 - SEND SECONDARY ADDRESS FOR LISTEN

AX = 4005h
BL = channel number and mode (see #00355)
Return: CF clear if successful
CF set on error
AL = error number (see #00354)
SeeAlso: AX=4004h,AX=4006h

Bitfields for I1541 channel number and mode:

Bit(s) Description (Table 00355)

7-4 mode
0110 read/write
1110 close channel
1111 open channel
3-0 channel number

-----t-144006-----
INT 14 - I1541 - SEND UNLISTEN SIGNAL

AX = 4006h
Return: CF clear if successful
CF set on error
AL = error number (see #00354)
SeeAlso: AX=4004h,AX=4005h

-----t-144007-----
INT 14 - I1541 - SEND TALK SIGNAL
AX = 4007h

BH = device number (0-15)
Return: CF clear if successful
CF set on error
AL = error number (see #00354)
SeeAlso: AX=4008h,AX=4009h
-----t-144008-----
INT 14 - I1541 - SEND SECONDARY ADDRESS FOR TALK
AX = 4008h
BL = channel number and mode (see #00355)
Return: CF clear if successful
CF set on error
AL = error number (see also #00354)
40h turn around time-out
SeeAlso: AX=4007h,AX=4009h
-----t-144009-----
INT 14 - I1541 - SEND UNTALK SIGNAL
AX = 4009h
Return: CF clear if successful
CF set on error
AL = error number (see #00354)
SeeAlso: AX=4007h,AX=4008h
-----t-14400A-----
INT 14 - I1541 - SEND A BYTE TO A DEVICE
AX = 400Ah
BL = byte to send
CL = last-byte flag
00h more bytes follow
01h this is the last byte to be sent
Return: CF clear if successful
CF set on error
AL = error number (see #00354)
SeeAlso: AX=4000h,AX=400Bh
-----t-14400B-----
INT 14 - I1541 - RECEIVE A BYTE FROM A DEVICE
AX = 400Bh
Return: CF clear if successful
AL = byte received
CL = last-byte flag
00h more bytes to follow
01h received byte is the last
CF set on error

```
    AL = error number
    80h EOI response required
    81h talker not ready
    82h clock not set
    83h clock not released
    CL = 00h
SeeAlso: AX=4000h,AX=400Ah
-----t-14400C-----
INT 14 - I1541 - WAIT
    AX = 400Ch
    CX = number of 838ns microticks to wait (0000h means 65536, ~55ms)
Return: after wait period elapses
SeeAlso: AX=4000h,AX=400Dh
-----t-14400D-----
INT 14 - I1541 - LONGWAIT
    AX = 400Dh
    DX:CX = number of 838ns microticks to wait
            (0000h:0000h means 4294967296, about one hour)
Example: To wait 1s you must set DX:CX=(1s/838ns)=1193180
SeeAlso: AX=4000h,AX=400Ch
-----t-14400E-----
INT 14 - I1541 - GET INFO
    AX = 400Eh
Return: AX = LPT port I/O address in use (0000h if no cable in use)
    BL = LPT number (1..3) in use (00h if no cable in use)
    CF set if the cable is auto-detectable
    CF clear if cable could not be auto-detected or is not present
SeeAlso: AX=4000h
-----S-1456-----
INT 14 U - BWC0M14 - INSTALLATION CHECK
    AH = 56h
Return: CX = 0001h if installed
Program: BWC0M14 is a network serial port emulator (simulating a Hayes modem
        connected to the serial port) distributed as part of the
        Beame&Whiteside BW-NFS package
SeeAlso: AH=57h, AH=58h, INT 2F/AX=DF00h/BX=5445h
-----S-1457-----
INT 14 U - BWC0M14 - INITIALIZE
    AH = 57h
    DL = port number
Return: AL = initialization status (00h successful, 01h already initialized)
```

CX = port status (0001h port redirected, 0002h and FFFFh failed)
Note: after this call, all invocations of INT 14/AH=00h-03h for the specified
port will be handled by BWC0M14 until AH=58h is called
SeeAlso: AH=00h"SERIAL",AH=56h,AH=58h

-----S-1458-----

INT 14 U - BWC0M14 - SHUTDOWN
AH = 58h

Return: CX = status (0001h successful, 0002h not initialized)
Note: after this call, BWC0M14 will no longer redirect the COM port
SeeAlso: AH=56h,AH=57h

-----N-146F--BXFFFE-----

INT 14 U - Connection Manager - ???
AH = 6Fh
BX = FFFFEh
???

Return: ???
Program: Connection Manager by Softwarehouse Corp. permits the sharing of
serial ports over an IPX or NetBIOS-based network

-----N-146F--BXFFFF-----

INT 14 - Connection Manager - INSTALLATION CHECK
AH = 6Fh
BX = FFFFh

Return: DX:BX -> Connection Manager Communication Table if installed
BX = FFFFh if not installed
SeeAlso: AH=0Dh/DX=FFFFh

-----S-146F00-----

INT 14 - HP Vectra EX-BIOS - "F14_INQUIRE" - INSTALLATION CHECK
AX = 6F00h
BX <> 4850h (usually set to 0000h for simplicity)
Return: BX = 4850h ("HP") if HP Extended BIOS serial port extensions available
AX destroyed

Note: supported by original HP Vectra AT and by ES/QS/RS series Vectras
SeeAlso: AX=6F01h,AX=6F02h,AX=6F03h,AX=6F04h,INT 10/AX=6F00h,INT 14/AX=6F00h
SeeAlso: INT 17/AX=6F00h,INT 33/AX=6F00h

-----S-146F01-----

INT 14 - HP Vectra EX-BIOS - "F14_EXINIT" - INITIALIZE SERIAL PORT
AX = 6F01h
BX = port attributes (see #00356)
DX = port number (0-3)

Return: AH = line status (see #00304)
AL = modem status (see #00305)

Note: supported by original HP Vectra AT and by ES/QS/RS series Vectras

SeeAlso: AX=6F00h

Bitfields for HP Vectra Extended BIOS serial port attributes:

Bit(s) Description (Table 00356)

8-5 data rate (110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200)

4-3 parity

00 none

01 odd

10 none

11 even

2 stop bits (0 = one, 1 = two)

1-0 bits per character

10 seven-bit characters

11 eight-bit characters

0x undefined

-----S-146F02-----

INT 14 - HP Vectra EX-BIOS - "F14_PUT_BUFFER" - TRANSMIT BUFFER

AX = 6F02h

CX = number of characters in buffer

DX = port number (0-3)

ES:DI -> buffer containing characters

Return: AH = line status (see #00304)

AL = modem status (see #00305)

CX = number of bytes actually sent

ES:DI -> next byte to be transferred (unchanged if all bytes sent)

Desc: send characters from the specified buffer until all characters have
been sent or an error/timeout is encountered

Note: supported by original HP Vectra AT and by ES/QS/RS series Vectras

SeeAlso: AX=6F00h,AX=6F03h,AX=6F04h,INT 17/AX=6F02h

-----S-146F03-----

INT 14 - HP Vectra EX-BIOS - "F14_GET_BUFFER" - READ DATA INTO BUFFER

AX = 6F03h

CX = size of buffer

DX = port number (0-3)

ES:DI -> buffer for received characters

Return: AH = line status (see #00304)

---on error (AH bit 7 set)---

AL = 00h

ES:DI -> next byte to be transferred

---if successful---

AL = last byte read
ES:DI unchanged
CX = number of bytes read
Desc: read characters into the specified buffer until the buffer is filled
or a timeout occurs
Notes: supported by original HP Vectra AT and by ES/QS/RS series Vectras
polls the Data Set Ready modem status and Data Ready line status bits
to determine when characters are available
SeeAlso: AX=6F00h,AX=6F02h,AX=6F04h

-----S-146F04-----

INT 14 - HP Vectra EX-BIOS - "F14_TRM_BUFFER" - READ UNTIL TERMINATOR

AX = 6F04h
BL = lowest termination character
BH = highest termination character
CX = size of buffer
DX = port number (0-3)
ES:DI -> buffer for received characters

Return: AH = line status (see #00304)

---on error (AH bit 7 set)---

AL = 00h

ES:DI -> next byte to be transferred

---if successful---

AL = last byte read

ES:DI unchanged

CX = number of bytes read

Desc: read characters into the specified buffer until the buffer is filled,
a character in the specified range is received, or a timeout occurs

Notes: supported by original HP Vectra AT and by ES/QS/RS series Vectras
polls the Data Set Ready modem status and Data Ready line status bits
to determine when characters are available

SeeAlso: AX=6F00h,AX=6F02h,AX=6F03h

-----U-147000-----

INT 14 - NEWCOM - INSTALLATION CHECK

AX = 7000h

Return: AX = 4E43h ('NC') if installed

BX = 4F4Dh ('OM') if installed

CH = major version number

CL = minor version number

DH = patch level

DL = language (currently: 00h English, 01h French)

Program: Newcom is a DOS commandline-enhancer by kilobug@kali.isicom.fr

SeeAlso: INT 2F/AX=D44Dh

-----S-147E-----

INT 14 - FOSSIL - INSTALL AN EXTERNAL APPLICATION FUNCTION

AH = 7Eh

AL = code assigned to external application (80h-BFh)

80h reserved for communications FOSSIL

81h video FOSSIL

82h reserved for keyboard FOSSIL

83h reserved for system FOSSIL

ES:DX -> entry point

Return: AX = 1954h

BL = code assigned to application (same as input AL)

DH = status

00h failed

01h successful

SeeAlso: AH=7Fh,AH=80h"FOSSIL",AX=8100h,AH=82h"FOSSIL",AH=83h"FOSSIL"

-----S-147F-----

INT 14 - FOSSIL - REMOVE AN EXTERNAL APPLICATION FUNCTION

AH = 7Fh

AL = code assigned to external application

ES:DX -> entry point

Return: AX = 1954h

BL = code assigned to application (same as input AL)

DH = status

00h failed

01h successful

SeeAlso: AH=7Eh

-----S-1480-----

INT 14 - COMMUNICATIONS FOSSIL

AH = 80h

SeeAlso: AH=7Eh

-----S-1480-----

INT 14 - COURIERS.COM - INSTALLATION CHECK

AH = 80h

Return: AH = E8h if loaded

Program: COURIERS is a TSR utility by PC Magazine

-----S-148000-----

INT 14 - ARTICOM - INSTALLATION CHECK

AX = 8000h

Return: AL = FFh if installed

BH = major version

BL = minor version

Program: ArtiCom is an asynchronous communications driver by Artisoft which works on top of NetBIOS and allows modem/serial-port sharing by programs using INT 14 for serial I/O.

Note: ArtiCom supports 32 simultaneous COM ports using multiport cards and drivers

SeeAlso: AH=00h"SERIAL",AH=01h,AH=02h,AH=03h,AH=04h"SERIAL",AH=05h"SERIAL"

SeeAlso: AX=8001h,AX=8002h

-----S-148000-----

INT 14 - COMM-DRV v14.0 - READ PORT METRICS - GET ERROR CODE AND BUFFER STATUS

AX = 8000h

DX = port number

Return: AX = code for last error (see #00357)

BX = number of characters in output buffer

CX = nubmer of characters in input buffer

DX = state flag (see #00358)

Program: COMM-DRV is a universal serial communications driver by Willies'

Computer Software Company, which supports standard INT 14 and

FOSSIL calls as well as its own interfaces

SeeAlso: AX=8001h"COMM-DRV",AX=8002h"COMM-DRV",AX=8003h"COMM-DRV"

(Table 00357)

Values for COMM-DRV error code:

00h no error

01h buffer not set or attempted to change buffer for active port

02h port not active

03h transmit buffer full

04h receive buffer full

05h syntax error

06h invalid buffer size

07h invalid port

08h handler changed

09h invalid baud rate

0Ah invalid parity setting

0Bh invalid data length

0Ch invalid number of stop bits

0Dh invalid protocol number

0Eh IRQ changed

0Fh port changed

10h invalid threshold setting

11h invalid IRQ number

12h interrupts not enabled
13h invalid break syntax
14h fatal error
15h CTS error
16h invalid RS232 I/O port address
17h environment variable not set
18h error on IOCTL call
19h error during atexit cleanup
1Ah error mapping for direct calls
1Bh error opening device
1Ch unable to allocate memory
1Dh error on external micro card
1Eh card changed error
1Fh card type error
20h not supported
21h parent port error
22h card command buffer full
23h no subdevice for this port
24h unknown error
25h external card busy
26h no more timers available
27h INT 14 vector changed
28h INT 08 vector changed
29h DPMI error
2Ah TSR buffer too small (or nonexistent)
2Bh out of asynchronous resources
2Ch out of timer resources
2Dh out of "other" timer resources
2Eh file I/O error
2Fh hardware memory > 64K

Bitfields for state flag :

Bit(s) Description (Table 00358)

0 port is active
1 output throttled (XOFF received, or DSR or CTS reset)
2 input throttled (XOFF sent, or DTR or RTS reset)

-----S-148001-----

INT 14 - ARTICOM - UNLOAD ASYNCHRONOUS REDIRECTOR FROM MEMORY

AX = 8001h

Return: AX = error code, if error (see #00360)

SeeAlso: AX=8000h"ARTICOM",AX=8002h"ARTICOM",AX=8003h"ARTICOM"

Index: uninstall;ARTICOM

-----S-148001-----

INT 14 - COMM-DRV v14.0 - READ PORT METRICS - GET PORT PARAMETERS

AX = 8001h

DX = port number

Return: BX:DI -> Port Control Block (see #00367)

SeeAlso: AX=8000h"COMM-DRV",AX=8002h"COMM-DRV",AX=8003h"COMM-DRV"

-----S-148002-----

INT 14 - ARTICOM - GET ASYNCHRONOUS REDIRECTOR STATUS

AX = 8002h

ES:DI -> buffer for redirector status structure (see #00359)

Return: AX = error code, if error (see #00360)

SeeAlso: AX=8000h"ARTICOM",AX=8003h"ARTICOM"

Format of ARTICOM redirector status:

Offset Size Description (Table 00359)

00h WORD redirector major and minor version numbers

02h WORD redirectable ports found

04h WORD redirectable ports + local ports found

06h WORD redirector internal buffer size

08h WORD maximum servers maintained

0Ah WORD number of adapters found

-----S-148002-----

INT 14 - COMM-DRV v14.0 - READ PORT METRICS - GET PORT PARAMETERS

AX = 8002h

DX = port number

Return: AH bit 7 set on error

AH bit 7 clear if successful

BX:DI -> Port Control Block (see #00367) (modifyable portion only)

SeeAlso: AX=8000h"COMM-DRV",AX=8001h"COMM-DRV",AX=8003h"COMM-DRV"

-----S-148003-----

INT 14 - ARTICOM - TRANSLATE ERROR CODE TO ERROR STRING

AX = 8003h

CX = error number to translate (see #00360)

Return: ES:DI -> ASCIZ error text or 0000h:0000h if unable to translate

SeeAlso: AX=8000h

(Table 00360)

Values for ARTICOM error codes:

00h "No error"

01h "An invalid port number was specified"

```

02h "Port is already redirected"
03h "Too many ports redirected"
04h "Cannot locate the server"
05h "Server is busy"
06h "Access denied"
07h "Resource in use"
08h "Resource in use - request queued"
09h "No such resource"
0Ah "Invalid username/password pair"
0Bh "Noncompatible version number"
0Ch "Can't remove from memory"
0Dh "Bad NETBIOS adapter number"
0Eh "No more entries in list"
0Fh "Resource is not available at this time"
10h "Invalid value to INT 14 call"

```

-----S-148003-----

INT 14 - COMM-DRV v14.0 - READ PORT METRICS - GET I/O BUFFER SIZES

AX = 8003h

DX = port number

Return: AX = number of characters in input buffer

BX = input buffer size

CX = number of characters in output buffer

DX = output buffer size

SeeAlso: AX=8000h"COMM-DRV",AX=8001h"COMM-DRV",AX=8002h"COMM-DRV"

-----S-148004-----

INT 14 - ARTICOM - ATTACH ASYNCHRONOUS RESOURCE

AX = 8004h

DX = port to redirect (COM1=0, COM2=1, ...)

CH = attach type

CL = adapter to use for attach, 0FFh to search all

ES:DI -> attachment structure (see #00361)

Return: AX = error code, if error (see #00360)

Note: The wildcard '*' is supported in the server and resource fields. If

wild cards are used then the first matching available server is attached.

SeeAlso: AX=8000h,AX=8003h,AX=8005h

Format of ARTICOM attachment structure:

Offset Size Description (Table 00361)

00h 16 BYTES server to look for attach

10h 16 BYTES attach to resource name

20h 16 BYTES username for attach
30h 16 BYTES password for username or resource
40h BYTE attach type
00h normal
01h queue if resource is in use (not yet supported in v1.00)

-----S-148005-----

INT 14 - ARTICOM - DETACH ASYNCHRONOUS RESOURCE

AX = 8005h

DX = port to detach (COM1=0, COM2=1, ...)

Return: AX = error code, if error (see #00360)

Note: only a previously attached resource can be detached

SeeAlso: AX=8000h,AX=8003h,AX=8004h

-----S-148006-----

INT 14 - ARTICOM - GET RESOURCE INFORMATION

AX = 8006h

BX = remote port (COM1=0, COM2=1, ...)

CL = adapter number, FFh to try all adapters

ES:DI -> resource information structure (see #00362)

DS:SI -> 16 byte server name. See note.

Return: AX = error code, if error (see #00360)

BX = next remote port, recall to get next resource info

Note: Wild cards supported in both the resource field and server name string DS:SI. If wild cards used then first matching available resource information is searched. Set the resource field to FFh to return all resources.

SeeAlso: AX=8000h,AX=8002h,AX=8003h,AX=8007h

Format of ARTICOM resource information structure:

Offset Size Description (Table 00362)

00h BYTE 00h = free, else used

01h 16 BYTES resource name

11h 16 BYTES username of resource user

21h WORD amount of time used

23h WORD amount of time remaining

53h 48 BYTES description of resource

93h 64 BYTES initialization string for modem

B3h 32 BYTES dial string for modem

D3h 32 BYTES hang-up string for modem

-----S-148007-----

INT 14 - ARTICOM - GET REDIRECTED PORT INFORMATION

AX = 8007h

DX = port index (COM1=0, COM2=1, ...)
ES:DI -> buffer for port information structure (see #00363)
Return: CF clear if redirection info returned and port is redirected
CF set if not a redirected port
AX = error code, if error (see #00360)
SeeAlso: AX=8000h,AX=8003h,AX=8006h,AX=8008h

Format of ARTICOM port information structure:

Offset	Size	Description (Table 00363)
00h	16 BYTES	server name resource is on
10h	BYTE	adapter number server is on
11h	16 BYTES	resource name
21h	WORD	remote port index, use to get additional information
23h	WORD	buffer size
25h	WORD	baud rate (see #00364)
26h	BYTE	modem status register
27h	BYTE	modem control register
28h	BYTE	line status register
29h	BYTE	line control register
2Ah	BYTE	flow control in use: 0 - NONE, 1 - XON/XOFF, 2 - RTS/CTS
2Bh	WORD	send timeout in ticks
2Dh	WORD	receive timeout in ticks
2Fh	WORD	time used on remote port
31h	WORD	time left before timeout
33h	BYTE	if server changes allowed?
34h	WORD	FFFFh (-1) if connection ok, else old port index

(Table 00364)

Values for ARTICOM baud rate:

00h	110
01h	150
02h	300
03h	600
04h	1200
05h	2400
06h	4800
07h	9600
08h	19200
09h	38400
0Ah	57600
0Bh	115200

0Ch 134.5
0Dh 1800
0Eh 2000
0Fh 3600
10h 7200

SeeAlso: #00309

-----S-148008-----

INT 14 - ARTICOM - GET AVAILABLE SERVER NAME

AX = 8008h

BX = server index (0,1,...)

ES:DI -> server name structure (see #00365)

Return: AX = error code, if error (see #00360)

BX = next remote port, repeat call to get next available server

Note: the wildcard '*' is supported in the server name field. Set the

server name to FFh to search for all servers.

SeeAlso: AX=8000h,AX=8003h,AX=8007h

Format of ARTICOM server name structure:

Offset Size Description (Table 00365)

00h 16 BYTES (call) ASCIZ server name

10h BYTE (ret) the adapter server is found

-----S-148009-----

INT 14 - ARTICOM - SET SEND AND RECEIVE TIMEOUTS

AX = 8009h

BX = send timeout in ticks

CX = receive timeout in ticks

DX = port index (COM1=0, COM2=1, ...)

Return: nothing

SeeAlso: AX=8000h,AX=800Ah

-----S-14800A-----

INT 14 - ARTICOM - MODIFY FLOW CONTROL

AX = 800Ah

BL = flow control type (00h none, 01h XON/XOFF, 02h RTS/CTS)

DX = port index (COM1=0, COM2=1, ...)

Return: AX = error code, if error (see #00360)

Note: for attached ports only!

SeeAlso: AX=8000h,AX=8003h,AX=8009h

-----S-148025-----

INT 14 - ARTICOM - SET INTERNAL SEND/RECEIVE VECTOR

AX = 8025h

DS:DX -> address of trap function (see #00366) to call on read/write

Return: nothing

Note: setting the vector to a user function allows the redirector's activity to be monitored.

SeeAlso: AX=8000h,AX=8035h,INT 21/AH=25h

(Table 00366)

Values ARTICOM trap function is called with:

AH = operation

80h reading character

81h writing character

AL = character

Return: AX must be preserved

far JUMP to old trap function (see AX=8035h)

-----S-148035-----

INT 14 - ARTICOM - GET INTERNAL SEND/RECEIVE VECTOR

AX = 8035h

Return: ES:BX -> address of current send/receive routine

Note: this function returns the address of the routine which is called inside A-REDIR.EXE each time a character is received or sent on the active COM port.

SeeAlso: AX=8000h,AX=8025h,INT 21/AH=35h

-----S-1481-----

INT 14 - COURIERS.COM - CHECK IF PORT BUSY

AH = 81h

AL = port number (1-4)

Return: AH = status

00h port available

01h port exists but already in use

02h port nonexistent

Program: COURIERS is a TSR utility by PC Magazine

SeeAlso: AH=83h, AH=8Dh

-----S-1481-----

INT 14 - COMM-DRV - EXTENDED INITIALIZATION

AH = 81h

BX:DI -> port control block (see #00367)

DX = port number

Return: AH = line status register (see #00304)

error if bit 7 set

AL = modem status register (see #00305)

Program: COMM-DRV is a universal serial communications driver by Willies'

Computer Software Company, which supports standard INT 14 and

FOSSIL calls as well as its own interfaces

Note: AX=8001h should be called first to fill in the port control block

SeeAlso: AH=00h,AX=8001h,AH=82h"COMM-DRV",AH=86h"COMM-DRV"

Format of COMM-DRV port control block:

Offset Type Description (Table 00367)

00h	WORD	port IO address
02h	WORD	port IRQ
04h	WORD	baud rate
06h	WORD	parity
08h	WORD	data bits
0Ah	WORD	stop bits
0Ch	WORD	break status (0000h off)
0Eh	WORD	flow control protocol
10h	BYTE	input block
11h	BYTE	output block
12h	WORD	low threshold
14h	WORD	high threshold
16h	WORD	segment of buffer
18h	WORD	offset of buffer
1Ah	WORD	input buffer length
1Ch	WORD	output buffer length
1Eh	BYTE	auxiliary address
1Fh	BYTE	spare
20h	4 WORDs	spares

-----V-148100-----

INT 14 - VIDEO FOSSIL - RETURN VFOSSIL INFORMATION

AX = 8100h

ES:DI -> buffer for VFOSSIL information (see #00368)

Return: AX = 1954h if installed

SeeAlso: AH=7Eh,AX=8101h

Format of VFOSSIL information:

Offset Size Description (Table 00368)

00h	WORD	size of information in bytes, including this field
02h	WORD	VFOSSIL major version
04h	WORD	VFOSSIL revision level
06h	WORD	highest VFOSSIL application function supported

-----V-148101-----

INT 14 - VIDEO FOSSIL - OPEN VFOSSIL

AX = 8101h

ES:DI -> buffer for application function table (see #00369)

CX = length of buffer in bytes

Return: AX = 1954h if installed

BH = highest VFOSSIL application function supported

Note: the number of initialized pointers in the application function table will never exceed CX/4; if the buffer is large enough, BH+1 pointers will be initialized

SeeAlso: AX=8102h

Format of VFOSSIL application function table:

Offset Size Description (Table 00369)

00h	DWORD	-> function to query current video mode (VioGetMode) (see #00374)
04h	DWORD	-> function to set video mode (VioSetMode) (see #00375)
08h	DWORD	-> function to query hardware config (VioGetConfig) (see #00376)
0Ch	DWORD	-> function to write data in TTY mode (VioWrtTTY) (see #00377)
10h	DWORD	-> function to get current ANSI state (VioGetANSI) (see #00378)
14h	DWORD	-> function to set new ANSI state (VioSetANSI) (see #00379)
18h	DWORD	-> function to get curr cursor position (VioGetCurPos) (see #00380)
1Ch	DWORD	-> function to set cursor position (VioSetCurPos) (see #00381)
20h	DWORD	-> function to get cursor shape (VioGetCurType) (see #00382)
24h	DWORD	-> function to set cursor shape (VioSetCurType) (see #00383)
28h	DWORD	-> function to scroll screen up (VioScrollUp) (see #00384)
2Ch	DWORD	-> function to scroll screen down (VioScrollDn) (see #00385)
30h	DWORD	-> function to read cell string from screen (VioReadCellStr) (see #00386)
34h	DWORD	-> function to read char string from screen (VioReadCharStr) (see #00387)
38h	DWORD	-> function to write a cell string (VioWrtCellStr) (see #00388)
3Ch	DWORD	-> function to write char string, leaving attr (VioWrtCharStr) (see #00389)
40h	DWORD	-> function to write char string, const attr (VioWrtCharStrAttr) (see #00390)
44h	DWORD	-> function to replicate an attribute (VioWrtNAttr) (see #00391)
48h	DWORD	-> function to replicate a cell (VioWrtNCell) (see #00392)
4Ch	DWORD	-> function to replicate a character (VioWrtNChar) (see #00393)

Format of VFOSSIL video mode data structure:

Offset Size Description (Table 00370)

00h WORD length of structure including this field
02h BYTE mode characteristics
 bit 0: clear if MDA, set otherwise
 bit 1: graphics mode
 bit 2: color disabled (black-and-white)
03h BYTE number of colors supported (1=2 colors, 4=16 colors, etc)
04h WORD number of text columns
06h WORD number of text rows
08h WORD reserved
0Ah WORD reserved
0Ch DWORD reserved

SeeAlso: #00374, #00375

Format of VFOSSIL video configuration data:

Offset Size Description (Table 00371)

00h WORD structure length including this field
02h WORD adapter type
 00h monochrome/printer
 01h CGA
 02h EGA
 03h VGA
 07h 8514/A
04h WORD display type
 00h monochrome
 01h color
 02h enhanced color
 09h 8514
06h DWORD adapter memory size

SeeAlso: #00376

Format of VFOSSIL cursor type record:

Offset Size Description (Table 00372)

00h WORD cursor start line
02h WORD cursor end line
04h WORD cursor width (always 01h)
06h WORD cursor attribute (FFFFh = hidden)

(Table 00373)

Values for VFOSSIL error code:

0000h successful
0074h internal VIO failure
0163h unsupported mode
0166h invalid row value
0167h invalid column value
017Eh buffer too small
01A5h invalid VIO parameter
01B4h invalid VIO handle

(Table 00374)

Call VioGetMode with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to video mode data structure (see #00370)

Return: AX = error code (00h, 74h, 17Eh, 1B4h) (see #00373)

SeeAlso: #00375

(Table 00375)

Call VioSetMode with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to video mode data structure (see #00370)

Return: AX = error code (00h, 74h, 163h, 17Eh, 1A5h, 1B4h) (see #00373)

SeeAlso: #00374

(Table 00376)

Call VioGetConfig with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to video configuration data buffer (see #00371)

Return: AX = error code (00h, 74h, 17Eh, 1B4h) (see #00373)

(Table 00377)

Call VioWrtTTY with:

STACK: WORD VIO handle (must be 00h)

WORD length of string

DWORD pointer to character string to be written to screen

Return: AX = error code (00h, 74h, 1B4h) (see #00373)

Notes: write wraps at end of line and terminates if it reaches end of screen
in ANSI mode, ANSI control sequences are interpreted, and this func is
not required to be reentrant; in non-ANSI mode, the function is
reentrant and may be called from within an MS-DOS function call

(Table 00378)

Call VioGetANSI with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to WORD which will be set to 00h if ANSI is off
or 01h if ANSI is on

Return: AX = error code (00h, 74h, 1B4h) (see #00373)

SeeAlso: #00379

(Table 00379)

Call VioSetANSI with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to WORD indicating new state of ANSI
00h off, 01h on

Return: AX = error code (00h, 74h, 1A4h, 1B4h) (see #00373)

SeeAlso: #00378

(Table 00380)

Call VioGetCurPos with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to WORD to hold current cursor column (0-based)

DWORD pointer to WORD to hold current cursor row (0-based)

Return: AX = error code (00h, 74h, 1B4h) (see #00373)

SeeAlso: #00381

(Table 00381)

Call VioSetCurPos with:

STACK: WORD VIO handle (must be 00h)

WORD cursor column

WORD cursor row

Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)

Note: if either coordinate is invalid, the cursor is not moved

SeeAlso: #00380

(Table 00382)

Call VioGetCurType with:

STACK: WORD VIO handle (must be 00h)

DWORD pointer to cursor type record (see #00372)

Return: AX = error code (00h, 74h, 1B4h) (see #00373)

SeeAlso: #00383

(Table 00383)

Call VioSetCurType with:

STACK: WORD VIO handle (must be 00h)
DWORD pointer to cursor type record (see #00372)
Return: AX = error code (00h, 74h, 1A4h, 1B4h) (see #00373)
SeeAlso: #00384

(Table 00384)

Call VioScrollUp with:

STACK: WORD VIO handle (must be 00h)
DWORD pointer to char/attr cell for filling emptied rows
WORD number or rows to scroll (FFFFh = clear area)
WORD right column of scroll area
WORD bottom row of scroll area
WORD left column of scroll area
WORD top row of scroll area
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)
SeeAlso: #00385,INT 10/AH=06h

(Table 00385)

Call VioScrollDn with:

STACK: WORD VIO handle (must be 00h)
DWORD pointer to char/attr cell for filling emptied rows
WORD number or rows to scroll (FFFFh = clear area)
WORD right column of scroll area
WORD bottom row of scroll area
WORD left column of scroll area
WORD top row of scroll area
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)
SeeAlso: #00384,INT 10/AH=07h

(Table 00386)

Call VioReadCellStr with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start reading
WORD row at which to start reading
DWORD pointer to WORD containing length of buffer in bytes
on return, WORD contains number of bytes actually read
DWORD pointer to buffer for cell string
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)

(Table 00387)

Call VioReadCharStr with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start reading
WORD row at which to start reading
DWORD pointer to WORD containing length of buffer in bytes
on return, WORD contains number of bytes actually read
DWORD pointer to buffer for character string
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)

(Table 00388)

Call VioWrtCellStr with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start writing
WORD row at which to start writing
WORD length of cell string in bytes
DWORD pointer to cell string to write
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)

Note: write wraps at end of line and terminates if it reaches end of screen

(Table 00389)

Call VioWrtCharStr with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start writing
WORD row at which to start writing
WORD length of character string
DWORD pointer to character string to write
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)

Note: write wraps at end of line and terminates if it reaches end of screen

(Table 00390)

Call VioWrtCharStrAttr with:

STACK: WORD VIO handle (must be 00h)
DWORD pointer to attribute to be applied to each character
WORD column at which to start writing
WORD row at which to start writing
WORD length of character string
DWORD pointer to character string to write
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)

Note: write wraps at end of line and terminates if it reaches end of screen

(Table 00391)

Call VioWrtNAttr with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start writing
WORD row at which to start writing
WORD number of times to write attribute
DWORD pointer to display attribute to replicate
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)
Note: write wraps at end of line and terminates if it reaches end of screen

(Table 00392)

Call VioWrtNCell with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start writing
WORD row at which to start writing
WORD number of times to write cell
DWORD pointer to cell to replicate
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)
Note: write wraps at end of line and terminates if it reaches end of screen

(Table 00393)

Call VioWrtNChar with:

STACK: WORD VIO handle (must be 00h)
WORD column at which to start writing
WORD row at which to start writing
WORD number of times to write character
DWORD pointer to character to replicate
Return: AX = error code (00h, 74h, 166h, 167h, 1B4h) (see #00373)
Note: write wraps at end of line and terminates if it reaches end of screen

-----V-148102-----

INT 14 - VIDEO FOSSIL - CLOSE VFOSSIL

AX = 8102h

Return: AX = 1954h

Note: terminates all operations; after this call, the video FOSSIL may either
be removed from memory or reinitialized

SeeAlso: AX=8101h,AX=8103h

-----V-148103-----

INT 14 - VIDEO FOSSIL - UNINSTALL

AX = 8103h

Return: AX = 1954h

Note: this is an extension to the VFOSSIL spec by Bob Hartman's VFOS_IBM

-----K-1482-----

INT 14 - KEYBOARD FOSSIL

```
AH = 82h
SeeAlso: AH=7Eh
-----S-1482-----
INT 14 - COURIERS.COM - CONFIGURE PORT
  AH = 82h
  AL = port number (1-4)
  BX = speed (bps)
  CX = bit flags
    bit 0: enable input flow control
    bit 1: enable output flow control
    bit 2: use X.PC protocol (not yet implemented)
Return: nothing
SeeAlso: AH=00h,AH=8Ch,INT 7A"X.PC"
-----S-1482-----
INT 14 - COMM-DRV v14.0 - PORT CLEANUP
  AH = 82h
  DX = port number
Return: AH bit 7 set on error
  AH bit 7 clear if successful
Desc: reset the port to its state before the AH=81h initialization and unhook
  any interrupts used by the port
SeeAlso: AH=81h"COMM-DRV",AH=83h"COMM-DRV"
-----1483-----
INT 14 - SYSTEM FOSSIL
  AH = 83h
SeeAlso: AH=7Eh
-----S-1483-----
INT 14 - COURIERS.COM - START INPUT
  AH = 83h
  ES:BX -> circular input buffer
  CX = length of buffer
    (should be at least 128 bytes if input flow control enabled)
Return: nothing
SeeAlso: AH=18h,AH=87h,AH=8Dh,AH=A5h"BAPI"
-----S-1483-----
INT 14 - COMM-DRV v14.0 - FLUSH COMMUNICATION BUFFERS
  AH = 83h
  DX = port number
  AL = subfunction
    00h flush input buffer
    01h flush output buffer
```

```
    02h flush both buffers
Return: AH bit 7 set on error
    AH bit 7 clear if successful
SeeAlso: AH=81h"COMM-DRV",AH=84h"COMM-DRV"
-----S-1484-----
INT 14 - COURIERS.COM - READ CHARACTER
    AH = 84h
Return: ZF set if no characters available
    ZF clear
    AL = character
    AH = modem status bits
    bit 7: set on input buffer overflow
SeeAlso: AH=02h,AH=86h,AH=89h
-----S-1484-----
INT 14 - COMM-DRV v14.0 - SEND PACKET
    AH = 84h
    CX = packet length in bytes
    DX = port number
    ES:DI -> packet to be sent
Return: AH = line status (see #00304)
    bit 7 set on error
    AL destroyed
SeeAlso: AH=83h"COMM-DRV",AH=85h"COMM-DRV",AH=86h"COMM-DRV"
-----S-1485-----
INT 14 - COURIERS.COM - FLUSH PENDING INPUT
    AH = 85h
Return: nothing
SeeAlso: AH=0Ah,AH=88h"COURIERS"
-----S-1485-----
INT 14 - COMM-DRV v14.0 - RECEIVE PACKET
    AH = 85h
    CX = length of packet in bytes
    DX = port number
    ES:DI -> buffer for packet
Return: AH = line status (see #00304)
    bit 7 set on error
    AL destroyed
Note: this call requires that at least the requested number of bytes are
    already present in the input buffer, and will fail if there are
    fewer bytes available
SeeAlso: AH=84h"COMM-DRV",AH=86h"COMM-DRV",AH=8Eh"COMM-DRV"
```


-----S-1486-----

INT 14 - COURIERS.COM - START OUTPUT

AH = 86h

ES:BX -> output buffer

CX = length of output buffer

Return: nothing

SeeAlso: AH=19h,AH=83h"COURIERS",AH=A4h"BAPI"

-----S-1486-----

INT 14 - COMM-DRV v14.0 - SET INPUT/OUTPUT TIMEOUTS

AH = 86h

BL = maximum clock ticks to wait before signalling error on input func

BH = maximum clock ticks to wait before signalling error on output

DX = port number

SI = input timeout in clock ticks if BL=FFh and BH=FFh

DI = output timeout in clock ticks if BL=FFh and BH=FFh

Return: AH bit 7 set on error

AH bit 7 clear if successful

Note: functions 02h, 85h, and 8Eh will wait for the input timeout before returning an error when no data is available; functions 01h and 84h will wait for the output timeout before returning an error if there is no space to output the data

SeeAlso: AH=01h,AH=02h,AH=84h"COMM-DRV",AH=85h"COMM-DRV",AH=8Eh"COMM-DRV"

-----S-1487-----

INT 14 - COURIERS.COM - OUTPUT STATUS

AH = 87h

Return: AX = number of unsent characters

SeeAlso: AH=88h"COURIERS"

-----S-1487-----

INT 14 - COMM-DRV v14.0 - TURN ON DTR

AH = 87h

DX = port number

Return: AH bit 7 set on error

AH bit 7 clear if successful

SeeAlso: AX=8000h"COMM-DRV",AH=88h"COMM-DRV",AH=89h"COMM-DRV"

-----S-1488-----

INT 14 - COURIERS.COM - ABORT OUTPUT

AH = 88h

SeeAlso: AH=09h"FOSSIL",AH=85h"COURIERS"

-----S-1488-----

INT 14 - COMM-DRV v14.0 - TURN OFF DTR

AH = 88h

```
DX = port number
Return: AH bit 7 set on error
AH bit 7 clear if successful
Program: COMM-DRV is a universal serial communications driver by Willies'
Computer Software Company, which supports standard INT 14 and
FOSSIL calls as well as its own interfaces
SeeAlso: AX=8000h"COMM-DRV",AH=87h"COMM-DRV",AH=8Ah"COMM-DRV"
-----S-1489-----
INT 14 - COURIERS.COM - SEND SINGLE CHARACTER
AH = 89h
CL = character to send
Return: nothing
SeeAlso: AH=01h,AH=84h"COURIERS"
-----S-1489-----
INT 14 - COMM-DRV v14.0 - TURN ON RTS
AH = 89h
DX = port number
Return: AH bit 7 set on error
AH bit 7 clear if successful
SeeAlso: AX=8000h"COMM-DRV",AH=87h"COMM-DRV",AH=8Ah"COMM-DRV"
-----S-148A-----
INT 14 - COURIERS.COM - SEND BREAK
AH = 8Ah
Return: nothing
SeeAlso: AH=89h"COURIERS",AH=FAh
-----S-148A-----
INT 14 - COMM-DRV v14.0 - TURN OFF RTS
AH = 8Ah
DX = port number
Return: AH bit 7 set on error
AH bit 7 clear if successful
SeeAlso: AX=8000h"COMM-DRV",AH=88h"COMM-DRV",AH=89h"COMM-DRV"
-----S-148B-----
INT 14 - COMM-DRV v14.0 - SET USER INTERRUPT ROUTINE
AH = 8Bh
CX = bitmask of interrupt to process
00h = deinstall
BX:DI -> DWORD containing address of function to be called
Return: AH bit 7 clear if successful
AH bit 7 set on error
-----S-148C-----
```

INT 14 - COURIERS.COM - SET SPEED

AH = 8Ch

BX = speed in bps

Return: nothing

SeeAlso: AH=00h,AH=82h"COURIERS"

-----S-148C-----

INT 14 - COMM-DRV v14.0 - READ UART REGISTER

AH = 8Ch

AL = register offset

DX = port number

Return: AH bit 7 set on error

AH bit 7 clear if successful

AL = contents of UART register

SeeAlso: AH=8Dh"COMM-DRV"

-----S-148D-----

INT 14 - COURIERS.COM - DECONFIGURE PORT

AH = 8Dh

Return: nothing

SeeAlso: AH=82h"COURIERS"

-----S-148D-----

INT 14 - COMM-DRV v14.0 - WRITE UART REGISTER

AH = 8Dh

AL = register offset

BL = new value for UART register

DX = port number

Return: AH bit 7 set on error

AH bit 7 clear if successful

SeeAlso: AH=8Ch"COMM-DRV"

-----S-148E-----

INT 14 - COMM-DRV v14.0 - READ PACKET NONDESTRUCTIVELY

AH = 8Eh

CX = length of packet in bytes

DX = port number

ES:DI -> buffer for packet

Return: AH = line status (see #00304)

bit 7 set on error (see AX=8000h"COMM-DRV")

AL destroyed

Program: COMM-DRV is a universal serial communications driver by Willies'

Computer Software Company, which supports standard INT 14 and

FOSSIL calls as well as its own interfaces

Desc: retrieve a packet from the input buffer without removing it from the

buffer

Note: this call requires that at least the requested number of bytes are already present in the input buffer, and will fail if there are fewer bytes available

SeeAlso: AX=8000h"COMM-DRV",AH=84h"COMM-DRV",AH=85h"COMM-DRV",AH=86h"COMM-DRV"

-----S-14A0-----

INT 14 - 3com BAPI SERIAL I/O - CONNECT TO PORT

AH = A0h

ES:BX -> ASCIZ internet host name

CX = length of name

Return: AH = return code (00h,04h-06h,08h,0Ah-0Ch) (see #00394)

CL = session ID

Program: the Bridge Application Program Interface is a set of functions which makes many of the details of LAN communications transparent

Note: Novell TELAPI.EXE returns AH=09h (not supported) and CL=00h

SeeAlso: AH=A1h"BAPI",AH=A2h"BAPI",AH=A5h"BAPI",AX=AF00h

(Table 00394)

Values for 3com BAPI return code:

00h successful

01h no characters written

02h no characters read

03h no such session

04h clearinghouse name not found

05h no response from host

06h no more sessions available

07h session aborted

08h invalid clearinghouse name

09h not supported

0Ah internal (general) network error

0Bh out of memory

0Ch invalid IP address

-----S-14A0--CXFFFF-----

INT 14 - Interconnections Inc. TES - INSTALLATION CHECK/STATUS REPORT

AH = A0h

CX = FFFFh

Return: CF clear if successful

AX = 5445h ('TE')

CX <> FFFFh

DX = port number

CF set on error

Program: TES is a network serial port emulation program

SeeAlso: AH=A1h"TES"

-----S-14A1-----

INT 14 - 3com BAPI SERIAL I/O - DISCONNECT FROM PORT

AH = A1h

DH = session ID (00h for external session managment)

Return: AH = return code (00h,03h,07h,0Ah,0Bh) (see #00394)

AL destroyed (Novell TELAPI.EXE)

SeeAlso: AH=A0h"BAPI"

-----S-14A1-----

INT 14 - Interconnections Inc. TES - GET LIST OF SESSIONS WITH STATUS

AH = A1h

Return: CX = number of active sessions

ES:SI -> status array (see #00395)

SeeAlso: AH=A2h"TES",AH=A3h"TES"

Format of Interconnections TES status array entry:

Offset Size Description (Table 00395)

00h BYTE status

01h WORD offset of name

-----S-14A2-----

INT 14 - 3com BAPI SERIAL I/O - WRITE CHARACTER

AH = A2h

AL = character

DH = session ID (00h for external session managment)

Return: AH = return code (00h,01h,03h,07h,0Ah,0Bh) (see #00394)

SeeAlso: AH=A0h"BAPI",AH=A3h"BAPI",AH=A4h"BAPI"

-----S-14A2-----

INT 14 - Interconnections Inc. TES - GET LIST OF SERVER NAMES

AH = A2h

Return: CX = number of servers

ES:SI -> array of offsets from ES for server names

SeeAlso: AH=A1h"TES"

-----S-14A3-----

INT 14 - 3com BAPI SERIAL I/O - READ CHARACTER

AH = A3h

DH = session ID (00h for external session managment)

Return: AH = return code (00h,02h,03h,07h,0Ah,0Bh) (see #00394)

AL = character read or 00h if none available

SeeAlso: AH=A0h"BAPI",AH=A2h"BAPI",AH=A5h"BAPI",AH=A7h"BAPI"

-----S-14A3-----

INT 14 - Interconnections Inc. TES - START A NEW SESSION

AH = A3h

ES:SI -> ???

Return: CF clear if successful

AX = 5445h ('TE')

CX <> FFFFh

DX = port number

CF set on error

SeeAlso: AH=A1h"TES",AH=A4h"TES",AH=A6h"TES"

-----S-14A4-----

INT 14 - 3com BAPI SERIAL I/O - WRITE BLOCK

AH = A4h

CX = length of buffer in bytes

DH = session ID (00h for external session managment)

ES:BX -> buffer containing data

Return: AH = return code (00h,01h,03h,07h,0Ah,0Bh) (see #00394)

CX = number of bytes actually sent

SeeAlso: AH=19h, AH=86h, AH=A0h"BAPI", AH=A5h"BAPI"

-----S-14A4-----

INT 14 - Interconnections Inc. TES - HOLD CURRENTLY ACTIVE SESSION

AH = A4h

???

Return: ???

SeeAlso: AH=A3h"TES",AH=A5h"TES"

-----S-14A5-----

INT 14 - 3com BAPI SERIAL I/O - READ BLOCK

AH = A5h

CX = length of buffer

DH = session ID (00h for external session managment)

ES:BX -> buffer for data

Return: AH = return code (00h,02h,03h,07h,0Ah,0Bh) (see #00394)

CX = number of bytes actually read

SeeAlso: AH=18h, AH=83h"COURIERS", AH=A0h"BAPI", AH=A3h"BAPI", AH=A4h"BAPI"

SeeAlso: AH=A7h"BAPI", AX=FF02h

-----S-14A5-----

INT 14 - Interconnections Inc. TES - RESUME A SESSION

AH = A5h

AL = session number

Return: ???

SeeAlso: AH=A4h"TES",AH=A6h"TES"

-----S-14A6-----

INT 14 - 3com BAPI SERIAL I/O - SEND SHORT BREAK

AH = A6h

DH = session ID (00h for external session management)

Return: AH = return code (00h,03h,07h,0Ah,0Bh) (see #00394)

Desc: generate a short break signal; if data delivery was turned off by the
break, wait for the host to turn it on again

SeeAlso: AH=1Ah,AH=8Ah,AH=FAh,AH=A0h"BAPI"

-----S-14A6-----

INT 14 - Interconnections Inc. TES - DROP A SESSION

AH = A6h

AL = session number

Return: AH = status

00h successful

else error

SeeAlso: AH=A3h"TES",AH=A5h"TES"

-----S-14A7-----

INT 14 - 3com BAPI SERIAL I/O - READ STATUS

AH = A7h

DH = session ID (00h for external session management)

Return: AH = return code (00h,03h,07h,0Ah,0Bh) (see #00394)

CX = number of bytes available for reading

Note: Novell TELAPI.EXE v4.01 always returns either 0 or 1 bytes available

SeeAlso: AH=A5h"BAPI"

-----S-14A7-----

INT 14 - Interconnections Inc. TES - SWITCH TO NEXT ACTIVE SESSION

AH = A7h

???

Return: ???

SeeAlso: AH=A3h"TES",AH=A5h"TES"

-----S-14A8-----

INT 14 - Interconnections Inc. TES - SEND STRING TO COMMAND INTERPRETER

AH = A8h

AL = 00h no visible response

ES:SI -> ASCIZ command

Return: ???

-----N-14A8-----

INT 14 - Novell TelAPI v4.01 - CONNECTION INFORMATION???

AH = A8h

DH = session ID???

CH = subfunction

02h ???

```
    0Dh ???
    0Fh ???
    10h ???
    11h ???
    28h ???
    else
Return: AH = 09h (not supported)
Return: AH = return code (see #00394)
    00h successful
    CL = ??? (0/1/8) (subfunctions 02h,0Dh,0Fh,10h)
    CL = ??? (7Fh/FFh) (subfunction 28h)
    CX = ??? (subfunction 11h)
SeeAlso: AH=A9h"TelAPI"
-----N-14A9-----
INT 14 - Novell TelAPI v4.01 - CONNECTION CONTROL???
    AH = A9h
    DH = session ID???
    CH = subfunction
        02h ???
        0Dh ???
        0Fh ???
        10h ???
        11h ???
        28h ???
    else
Return: AH = 09h (not supported)
    ???
Return: AH = return code (see #00394)
    ???
SeeAlso: AH=A8h"TelAPI",AH=E4h,INT 6B/AX=0600h
-----V-14AA01-----
INT 14 - DimVGA v2.0+ - INSTALLATION CHECK
    AX = AA01h
Return: AX = FFFFh if installed, unchanged
    BX = version (v1.5+ only), BH = major, BL = minor (v1.5 = 0105h)
    CX = resident segment (v3.1+)
Program: DimVGA is a public domain screen saver by Menno Pieters
SeeAlso: AX=AA02h,AX=AA03h,AX=AA06h,INT 11/AX=0225h/BX=6900h,INT 12"KEYBUI"
SeeAlso: INT 2D/AL=10h"Burnout Plus",INT 2F/AX=6400h,INT 2F/AH=93h
SeeAlso: INT 2F/AX=C000h"VGAsave",INT 2F/AX=C000h"AD-DOS",INT 2F/AX=C050h
SeeAlso: INT 2F/AX=E300h
```


Index: screen saver;DimVGA

-----V-14AA02-----

INT 14 - DimVGA v2.0+ - SET TIME-OUT (DIMMING/BLANKING) PERIOD

AX = AA02h

BX = number of clock ticks

Return: AX = FFFFh

Note: on screen modes with 256 or less colors DimVGA will dim the screen,
when more than 256 colors can be used DimVGA will blank the screen.

SeeAlso: AX=AA01h,AX=AA03h,AX=AA04h,AX=AA06h

Index: screen saver;DimVGA

-----V-14AA03-----

INT 14 - DimVGA v2.0+ - SET DIMMING FACTOR

AX = AA03h

BX = percentage remaining visible (1-99)

Return: AX = FFFFh

SeeAlso: AX=AA02h,AX=AA05h,AX=AA06h

Index: screen saver;DimVGA

-----V-14AA04-----

INT 14 - DimVGA v2.0+ - GET TIME-OUT PERIOD

AX = AA04h

Return: AX = FFFFh

BX = current time-out in clock ticks

SeeAlso: AX=AA02h,AX=AA05h,AX=AA0Ah

Index: screen saver;DimVGA

-----V-14AA05-----

INT 14 - DimVGA v2.0+ - GET DIMMING FACTOR

AX = AA05h

Return: AX = FFFFh

BX = current dimming factor

SeeAlso: AX=AA03h,AX=AA04h,AX=AA0Ah

Index: screen saver;DimVGA

-----V-14AA06-----

INT 14 - DimVGA v2.0+ - DISABLE

AX = AA06h

Return: AX = FFFFh

SeeAlso: AX=AA01h,AX=AA07h,AX=AA0Ah

Index: screen saver;DimVGA

-----V-14AA07-----

INT 14 - DimVGA v2.0+ - ENABLE

AX = AA07h

Return: AX = FFFFh

SeeAlso: AX=AA01h,AX=AA06h,AX=AA0Ah

Index: screen saver;DimVGA

-----V-14AA08-----

INT 14 - DimVGA v2.0+ - DIM SCREEN 'MANUALLY'

AX = AA08h

Return: AX = FFFFh

Note: this function will dim the screen immediately, even if DimVGA is currently disabled

SeeAlso: AX=AA01h,AX=AA02h,AX=AA09h

Index: screen saver;DimVGA

-----V-14AA09-----

INT 14 - DimVGA v2.0+ - UNDIM SCREEN 'MANUALLY'

AX = AA09h

Return: AX = FFFFh

Note: this function will undim the screen immediately, even if DimVGA is currently disabled

SeeAlso: AX=AA01h,AX=AA08h

Index: screen saver;DimVGA

-----V-14AA0A-----

INT 14 - DimVGA v2.0+ - CHECK WHETHER ENABLED

AX = AA0Ah

Return: AX = FFFFh

BX = current state (0000h disabled, 0001h enabled)

SeeAlso: AX=AA01h,AX=AA06h,AX=AA07h

Index: screen saver;DimVGA

-----V-14AA0B-----

INT 14 - DimVGA v2.1+ - SET HOTKEY

AX = AA0Bh

BH = shift state (see #00396)

BL = keyboard scancode

Return: AX = FFFFh

SeeAlso: AX=AA01h,AX=AA0Ch

Index: screen saver;DimVGA

Bitfields for DimVGA hotkey shift state:

Bit(s) Description (Table 00396)

7-4 unused

3 Alt key pressed

2 Ctrl key pressed

1 Left shift key pressed

0 Right shift key pressed

-----V-14AA0C-----

INT 14 - DimVGA v2.1+ - GET HOTKEY

AX = AA0Ch

Return: AX = FFFFh

BH = shift state (see #00396)

BL = keyboard scancode

SeeAlso: AX=AA01h,AX=AA0Bh

Index: screen saver;DimVGA

-----V-14AA0D-----

INT 14 - DimVGA v3.0+ - SET MOUSE CHECK STATUS

AX = AA0Dh

BX = new mouse check status

0000h mouse checking off

0001h mouse checking on

Return: AX = FFFFh

Note: before switching mouse checking on, a mouse driver should be found in memory. If no mouse driver is found, mouse checking should be switched off (resident DimVGA does not check by itself).

SeeAlso: AX=AA01h,AX=AA0Eh

Index: screen saver;DimVGA

-----V-14AA0E-----

INT 14 - DimVGA v3.0+ - GET MOUSE CHECK STATUS

AX = AA0Eh

Return: BX = mouse check status (0000h disabled, 0001h enabled)

SeeAlso: AX=AA01h,AX=AA0Dh

Index: screen saver;DimVGA

-----V-14AA0F-----

INT 14 - DimVGA v3.4 - SET LOCKING STATUS

AX = AA0Fh

BX = locking status

0000h disabled

0001h enabled

Return: AX = FFFFh

SeeAlso: AX=AA01h,AX=AA0Dh,AX=AA10h

Index: screen saver;DimVGA

-----V-14AA10-----

INT 14 - DimVGA v3.4 - GET MOUSE CHECK STATUS

AX = AA10h

Return: BX = locking status (0000h disabled, 0001h enabled)

SeeAlso: AX=AA01h,AX=AA0Dh,AX=AA0Fh

Index: screen saver;DimVGA

-----14AD-----

INT 14 - IBM SurePath BIOS - Officially "Private" Function

AH = ADh

SeeAlso: AH=AEh"IBM",AH=AFh"IBM"

-----14AE-----

INT 14 - IBM SurePath BIOS - Officially "Private" Function

AH = AEh

SeeAlso: AH=ADh"IBM",AH=AFh"IBM"

-----14AF-----

INT 14 - IBM SurePath BIOS - Officially "Private" Function

AH = AFh

SeeAlso: AH=ADh"IBM",AH=AEh"IBM"

-----S-14AF00BXAAAA-----

INT 14 - 3com BAPI SERIAL I/O - INSTALLATION CHECK

AX = AF00h

BX = AAAAh

Return: AX = AF01h if installed

BH = protocol type (if BX=AAAAh on entry)

01h NetManage TCP/IP

BL = version for protocol type (if BX=AAAAh on entry)

Note: early versions of the BAPI and the ROM BIOS simply destroy AX; this

behavior is used to determine whether the newer functions (AH=B0h,

AH=B1h,etc) are available

SeeAlso: AH=A0h"BAPI"

-----S-14B0-----

INT 14 - 3com BAPI SERIAL I/O - EN/DISABLE "ENTER COMMAND MODE" (ECM) CHARACTER

AH = B0h

AL = new state (00h disabled, 01h enabled)

Return: AH = return code (00h,07h,0Ah) (see #00394)

Note: disabling the ECM character allows applications to send data which

includes the ECM character

SeeAlso: AX=AF00h"BAPI",AH=B1h,AH=B2h

-----S-14B1-----

INT 14 - 3com BAPI SERIAL I/O - ENTER COMMAND MODE

AH = B1h

Return: AH = return code (00h,07h,0Ah) (see #00394)

Desc: provide a means for the application or terminal emulator to perform

the same action normally caused by the ECM character

SeeAlso: AH=B0h,AH=B2h

-----S-14B2-----

INT 14 - 3com BAPI SERIAL I/O - GET ECM WATCH STATE

AH = B2h
 Return: AH = return code (00h,07h,0Ah) (see #00394)
 AL = watch flag (00h disabled, 01h enabled)
 Desc: determine whether the ECM character is enabled
 SeeAlso: AH=B0h,AH=B1h

-----S-14B3-----
 INT 14 - 3com BAPI SERIAL I/O - GET/SET CONFIGURATION INFO
 AH = B3h
 AL = direction (00h get, 01h set)
 DH = session ID (00h for external session managment)
 DL = configuration item (00h = end-of-line mapping)
 CX = new configuration item value (if AL=01h)
 ---if DL=00h---
 CH = application EOL type (app to Telnet client)
 01h application will send lone CR
 02h application will send CR-? pair
 CL = driver EOL type (Telnet client to Telnet server)
 01h driver should send CR-NUL pair
 02h driver should send CR-LF pair

Return: AH = return code (00h,03h,09h-0Bh) (see #00394)
 ---if AL=00h---
 CX = configuration item value (above)
 SeeAlso: AH=B2h

-----N-14E0-----
 INT 14 - TelAPI - "telopen" - CREATE TELNET CONNECTION (BLOCKING)
 AH = E0h
 BX = port number to connect with (default 0017h used if <= 0)
 CX:DX = Internet address of remote host
 DS:DI -> 2-byte remote host (session) identifier
 ES:SI -> 1700-byte buffer for Telnet state record
 0000h:0000h to use TelAPI internally-allocated space
 Return: AX = status (0000h-0009h,FED3h,FF37h,FFBDh,FFC0h,FFCDh) (see #00397)
 ES:SI buffer filled with state record
 ES:SI -> internally-allocated state record in some versions
 Note: the remote host identifier may be used to refer to this connection
 SeeAlso: AH=E1h,AH=ECh,AX=FF00h

(Table 00397)

Values for TelAPI status:

0000h-7FFFh successful (session number)
 FED3h (-301) no session allocated, or out of TelAPI data space

```

FF37h (-201) all sessions in use
FFBDh (-67) unknown hostname
FFC0h (-64) host not functioning
FFC3h (-61) connection attempt refused
FFC4h (-60) connection attempt timed out
FFC8h (-56) socket already connected
FFCDh (-51) network is unreachable
FFDDh (-35) operation would block

```

-----S-14E000-----

INT 14 - MX5 Extended FOSSIL - GET MNP STATUS BLOCK

AX = E000h

DX = port number (0-3)

Return: ES:BX -> status block (see #00398)

Program: MX5 is a FOSSIL driver by MagicSoft which emulates MNP Level 5, and ships with the METZ terminal program as MTEMNP.DRV (a TSR despite the .DRV extension)

SeeAlso: AX=E006h

Format of MX5 Extended FOSSIL status block:

Offset Size Description (Table 00398)

```

00h BYTE flag: active (00h no, 01h yes)
01h BYTE MNP level (2,4,5)
02h BYTE series ID from remote MNP
03h DWORD total packets transmitted
07h DWORD duplicate packets transmitted
0Bh DWORD maximum speed
0Fh DWORD total packets received
13h DWORD duplicate packets received
17h DWORD maximum speed

```

-----S-14E001-----

INT 14 - MX5 Extended FOSSIL - GET/SET MNP LEVEL

AX = E001h

BH = function

00h get MNP level

01h set MNP level

BL = new level (00h none, 02h/04h/05h MNP level N)

DX = port number (0-3)

Return: BL = MNP level

SeeAlso: AX=E002h, AX=E003h, AX=E004h, AX=E006h

-----S-14E002-----

INT 14 - MX5 Extended FOSSIL - GET/SET MNP ANSWER/ORIGINATE MODE

AX = E002h
BH = function
 00h get answer/originate mode
 01h set mode
BL = new mode (00h originate [default], 01h answer)
DX = port number (0-3)

Return: BL = answer/originate mode

SeeAlso: AX=E001h,AX=E003h,AX=E006h

-----S-14E003-----

INT 14 - MX5 Extended FOSSIL - GET/SET MNP WAIT TICKS

AX = E003h
BH = function
 00h get wait ticks
 01h set wait ticks
BL = MNP wait ticks (default 0Eh)
DX = port number (0-3)

Return: BL = wait ticks

SeeAlso: AX=E001h,AX=E002h,AX=E006h

-----S-14E004-----

INT 14 - MX5 Extended FOSSIL - GET/SET MNP CONNECT SOUND LEVEL

AX = E004h
BH = function
 00h get sound level
 01h set sound level
BL = new sound level (00h off, 01h on [default])
DX = port number

Return: BL = sound state

Desc: specify whether MX5 should generate beeps after an MNP connection
 (three high beeps if successful, high then low on connection failure)

SeeAlso: AX=E002h,AX=E006h

-----S-14E005-----

INT 14 - MX5 Extended FOSSIL - UNINSTALL

AX = E005h
Return: BX = segment of MX5's memory block or 0000h on failure

Note: caller must free the returned memory block to complete the uninstall

SeeAlso: AX=E006h

-----S-14E006BX0000-----

INT 14 - MX5 Extended FOSSIL - INSTALLATION CHECK

AX = E006h
BX = 0000h

Return: BX = 4D58h ('MX') if installed

AH = major version

AL = minor version

SeeAlso: AX=E000h,AX=E001h,AX=E005h,AX=E007h

-----S-14E007-----

INT 14 - MX5 Extended FOSSIL - WAIT SPECIFIED NUMBER OF TICKS

AX = E007h

CX = number of ticks to wait

Return: nothing

SeeAlso: AX=E006h

-----N-14E1-----

INT 14 - TelAPI - "telclose" - TERMINATE TELNET CONNECTION

AH = E1h

BX = connection ID

Return: AX = status (0000h,FFF7h,maybe others) (see #00397)

Note: flushes and releases all buffers and data space used by the connection

SeeAlso: AH=E0h,AH=E6h,AX=FF00h

-----N-14E2-----

INT 14 - TelAPI - "telread" - BUFFERED READ

AH = E2h

BX = connection ID (see AH=E0h"TelAPI")

CX = length of buffer in bytes

ES:SI -> buffer for data

Return: AX > 0000h number of characters actually read

AX = 0000h host has closed connection

AX < 0000h error code (see #00397)

Note: translates CRLF into local EOL if the connection is in ASCII mode,
negotiates various Telnet options, and immediately executes several
different Telnet action commands

SeeAlso: AH=07h"TelAPI",AH=E3h,AH=E6h,AX=FF00h,INT 6B/AH=01h

-----N-14E3-----

INT 14 - TelAPI - "telwrite" - BUFFERED WRITE

AH = E3h

BX = connection ID

CX = length of buffer in bytes

ES:SI -> buffer containing data

Return: AX > 0000h number of characters actually written

AX < 0000h error code (see #00397)

Note: translates local EOL into CRLF if the connection is in ASCII mode,
sends the appropriate Telnet commands for the characters selected
for IP, AYT, AO, EC, EL, and Break

SeeAlso: AH=06h"TelAPI",AH=E2h,AH=E6h,AX=FF00h,INT 6B/AH=00h

-----N-14E4-----

INT 14 - TelAPI - "telioctl" - CONNECTION CONTROL

AH = E4h

BX = connection ID (see AH=E0h"TelAPI")

CX = Telnet command/option identifier (see #00400)

ES:SI -> buffer containing command/option argument (see #00399)

Return: AX = status (0000h, etc.) (see #00397)

Desc: start filter control, initiate Telnet option negotiation, or get filter
control status

SeeAlso: AH=A9h,AH=E6h,AX=FF00h,INT 6B/AX=0600h

Format of TelAPI Telnet command/option argument:

Offset Size Description (Table 00399)

00h 5 WORD numeric arguments

0Ah DWORD -> ASCIZ string

SeeAlso: #00400

(Table 00400)

Values for TelAPI Telnet command/option identifier:

01h ASCII args: none

02h BINARY args: none

03h LOCALECHO args: none client echos data

04h REMOTEECHO args: none server echos data

05h SGA args: none Suppress Go-Ahead signal

07h CHARMODE args: none no line-buffering

08h LINEMODE args: -> erase-line ch perform line-buffering

09h RECVEOL args: EOL type

0Ah SENDEOL args: EOL type

0Bh EOR args: none enable end-of-record sequence

0Dh BREAK args: -> break char

0Eh VERBOSE args: verbosity display Telnet negotiations?

0Fh AYT args: -> AYT escape ch

10h AO args: -> AO escape char

11h IP args: -> IP escape char

12h EC args: -> escape char

13h EL args: -> escape char

14h STATUS args: type; returns data in structure

18h TERMTYPE args: -> terminal type

19h ATTACHPORT args: port number ; returns session number

1Bh TRANSMIT_EOR args: EOR enabled append EOR to every telwrite?

SeeAlso: #00399

-----N-14E5-----

INT 14 - TelAPI - "telreset" - RESET ALL CONNECTIONS

AH = E5h

Return: AX = status (0000h,other) (see also #00397)

FFFFh unable to reset

Desc: close all sessions and reset TelAPI to defaults

SeeAlso: AH=E1h,AH=E6h,AX=FF00h

-----N-14E6-----

INT 14 - TelAPI - "telunload" - UNINSTALL

AH = E6h

Return: AX = status

0000h successful

FFFFh unable to uninstall

Notes: TelAPI also supports the NASI/NACS and NCSI APIs on INT 6B

this function invokes AH=E5h internally

SeeAlso: AH=E5h,AX=FF00h,INT 6B/AH=00h,INT 6B/AH=10h

-----N-14E7-----

INT 14 - TelAPI - "telist" - GET TELNET SESSION LIST

AH = E7h

ES:SI -> 10-word buffer for session list

Return: AX = 0000h (successful)

ES:SI buffer filled

Desc: determine, for each of the ten allowable sessions, whether the session is currently available

Note: each word in the buffer is filled with either 0000h to indicate that the corresponding session is unavailable, or 0001h if available

SeeAlso: AH=E0h,AH=E5h,AX=FF00h

-----N-14E8-----

INT 14 - TelAPI - "telattach" - ATTACH COM PORT TO/FROM TELNET SESSION

AH = E8h

BX = connection ID (see AH=E0h"TelAPI")

CX = serial port number (0000h-0003h = COM1-COM4)

Return: AX = status

0000h successful

FFFFh failed

SeeAlso: AH=E0h,AH=E9h,AX=FF00h

-----N-14E9-----

INT 14 - TelAPI - "telportasn" - GET SESSION NUMBER FOR COM PORT

AH = E9h

DX = serial port number (0000h-0003h = COM1-COM4)

Return: AX >= 0000h session number

AX < 0000h error code (see #00397)

SeeAlso: AH=E0h,AH=E8h,AH=EAh,AX=FF00h

-----N-14EA-----

INT 14 - TelAPI - "telstatus" - GET TELNET CONNECTION STATUS INFORMATION

AH = EAh

BX = connection ID (see AH=E0h"TelAPI")

ES:SI -> buffer for status info (see #00401)

Return: AX = status (0000h,FFFFh,etc.)

SeeAlso: AH=E9h,AH=EBh,AX=FF00h

Format of TelAPI Telnet connection status information:

Offset Size Description (Table 00401)

00h 4 BYTES remote host IP address

04h 20 BYTES reserved

18h WORD local port number

1Ah BYTE connection mode (00h = ASCII, 01h = Binary)

1Bh BYTE echo flag (00h local, 01h remote)

1Ch BYTE SGA flag (00h will, 01h won't)

1Dh BYTE EOR negotiation flag (00h do negotiate, 01h don't)

1Eh BYTE buffering (00h line mode, 01h character mode)

1Fh BYTE reserved

20h BYTE verbose flag (00h no, 01h verbose mode)

21h BYTE received EOL (00h no xlat, 01h CR, 02h LF, 03h CRLF)

22h BYTE sent EOL (00h no translation, 01h CR, 02h LF)

23h BYTE break character

24h BYTE IP escape character

25h BYTE AO escape character

26h BYTE AYT escape character

27h BYTE EC escape character

28h BYTE EL escape character

29h 41 BYTES ASCIZ Telnet-negotiated terminal type

52h 9 BYTES session ID

SeeAlso: #00402

-----N-14EB-----

INT 14 - TelAPI - "telname" - GET AVAILABLE/INUSE STATUS FOR ALL SESSIONS

AH = EBh

ES:SI -> buffer for session statuses (see #00402)

Return: ES:SI buffer filled

SeeAlso: AH=E9h,AH=EAh,AX=FF00h

Format of TelAPI session status information [array]:

```

Offset  Size  Description (Table 00402)
 00h  BYTE  session state (00h available, 01h connected)
 01h  9 BYTES session ID if connected
 0Ah  WORD  attached COM port if connected, FFFFh if not
SeeAlso: #00401
-----N-14EC-----
INT 14 - TelAPI - "telnblkopen" - CREATE TELNET CONNECTION (NON-BLOCKING)
  AH = ECh
  BX = port number to connect with (default 0017h used if <= 0)
  CX:DX = Internet address of remote host
  DS:DI -> 2-byte remote host (connection) identifier
  ES:SI -> 1700-byte buffer for Telnet state record
         0000h:0000h to use TelAPI internally-allocated space
Return: AX = status (0000h-0009h, FED3h, FF37h, FFBDh, FFC0h, FFCDh) (see #00397)
  ES:SI buffer filled with state record
  ES:SI -> internally-allocated state record in some versions
Notes:  the remote host identifier may be used to refer to this connection
        this function returns immediately; use AH=EDh to check whether the
        connection has been established yet
        this function is not supported by the Microdyne TelAPI v3.7
SeeAlso: AH=E0h"TelAPI",AH=EDh,AX=FF00h
-----N-14ED-----
INT 14 - TelAPI - "telpoll" - POLL TELNET SESSION FOR CONNECTION COMPLETION
  AH = EDh
  BX = connection ID (see AH=ECh)
Return: AX = status (0000h,0001h,FFFFh,etc.) (see also #00397)
        0000h session now connected
        0001h connection still in progress
Note:  this function is not supported by the Microdyne TelAPI v3.7
SeeAlso: AH=EDh,AX=FF00h
-----a-14F0F0-----
INT 14 - ASAP v1.0 - ???
  AX = F0F0h
  DX = ???
  ???
Return: ???
Program: ASAP (Automatic Screen Access Program) is a shareware screen reader
        by MicroTalk
SeeAlso: AX=F0F1h
-----a-14F0F1DX0000-----
INT 14 - ASAP v1.0 - INSTALLATION CHECK

```

```
AX = F0F1h
DX = 0000h
Return: DX = segment of resident code
        0000h if not installed
Program: ASAP (Automatic Screen Access Program) is a shareware screen reader
        by MicroTalk
```

```
SeeAlso: AX=F0F0h,INT 10/AX=3800h
```

```
-----S-14F4FF-----
```

```
INT 14 - IBM/Yale EBIOS SERIAL I/O - INSTALLATION CHECK
```

```
AX = F4FFh
DX = port (00h-03h)
Return: CF clear if present
```

```
AX = 0000h
CF set if not present
AX <> 0000h
```

```
SeeAlso: AH=36h"ComShare",AH=F9h,AH=FCh
```

```
-----S-14F9-----
```

```
INT 14 - IBM/Yale EBIOS SERIAL I/O - REGAIN CONTROL
```

```
AH = F9h
DX = port (00h-03h)
Return: nothing
```

```
SeeAlso: AX=F4FFh
```

```
-----S-14FA-----
```

```
INT 14 - IBM/Yale EBIOS SERIAL I/O - SEND BREAK
```

```
AH = FAh
DX = port (00h-03h)
Return: nothing
```

```
SeeAlso: AH=07h"MBBIOS",AH=1Ah,AH=8Ah
```

```
-----S-14FB-----
```

```
INT 14 - IBM/Yale EBIOS SERIAL I/O - SET OUTGOING MODEM SIGNALS
```

```
AH = FBh
AL = modem control register (see #00334 at AH=05h"SERIAL")
DX = port (00h-03h)
```

```
Return: nothing
```

```
SeeAlso: AH=05h"SERIAL"
```

```
-----S-14FC-----
```

```
INT 14 - IBM/Yale EBIOS SERIAL I/O - READ CHARACTER, NO WAIT
```

```
AH = FCh
DX = port (00h-03h)
Return: AH = RS232 status bits (see #00304 at AH=03h)
```

```
AL = character
```

SeeAlso: AH=02h,AH=0Ch,AX=FF02h

-----S-14FD02-----

INT 14 - IBM/Yale EBIOS SERIAL I/O - READ STATUS

AX = FD02h

Return: CX = number of characters available

-----N-14FF00-----

INT 14 - TelAPI - "telcheck" - INSTALLATION CHECK

AX = FF00h

Return: AX = 00FFh if installed

BX = version number * 100 (decimal)

SeeAlso: AH=E6h,AX=F4FFh

-----S-14FF01-----

INT 14 - IBM/Yale EBIOS SERIAL I/O - SET SEND BUFFER

AX = FF01h

CX = length of buffer (0000h to cancel buffer assignment)

DX = port (00h-03h)

ES:BX -> send buffer

Return: nothing

SeeAlso: AH=18h,AH=83h"COURIERS",AH=A5h"BAPI",AH=FCh,AX=FF02h

-----S-14FF02-----

INT 14 - IBM/Yale EBIOS SERIAL I/O - SET RECEIVE BUFFER

AX = FF02h

CX = length of buffer (0000h to cancel buffer assignment)

DX = port (00h-03h)

ES:BX -> receive buffer

Return: nothing

SeeAlso: AH=18h,AH=83h"COURIERS",AH=A5h"BAPI",AH=FCh,AX=FF01h

-----S-14FFF8-----

INT 14 - COMM-DRV v14.0 - SET BAUD RATE DIVISOR

AX = FFF8h

BX = card type (sub-device number)

CX = new baudrate divisor

DX = index to baud rate

Return: AH bit 7 set on error

AH bit 7 clear if successful

Program: COMM-DRV is a universal serial communications driver by Willies'

Computer Software Company, which supports standard INT 14 and

FOSSIL calls as well as its own interfaces

SeeAlso: AX=8000h"COMM-DRV"

-----S-14FFFB-----

INT 14 - COMM-DRV v14.0 - GET HIGHEST ALLOWED PORT NUMBER

AX = FFFBh
DX = port number
Return: AH bit 7 set on error
AH bit 7 clear if successful
BX = highest port number

-----S-14FFFC-----

INT 14 - COMM-DRV v14.0 - GET INT 14 FLAGS

AX = FFFCh
DX = port number
Return: AH bit 7 set on error
AH bit 7 clear if successful
BX = flags (see #00403)

SeeAlso: AX=FFFDh

Bitfields for INT 14h flags:

Bit(s) Description (Table 00403)

- 0 port active for INT 14h
- 1 interface behaving like a FOSSIL driver

-----S-14FFFD-----

INT 14 - COMM-DRV v14.0 - SET INT 14 FLAGS

AX = FFFDh
BX = flags (see #00403)
DX = port number
Return: AH bit 7 set on error
AH bit 7 clear if successful

SeeAlso: AX=FFFCh

-----S-14FFFE-----

INT 14 - COMM-DRV v14.0 - RESTORE INT 14 VECTOR TO ORIGINAL

AX = FFFEh
Return: AH bit 7 set on error
AH bit 7 clear if successful

-----S-14FFFF-----

INT 14 - COMM-DRV v14.0 - GET INT 14 INFORMATION AREA

AX = FFFFh
BX:SI -> DWORD buffer for address of information area (see #00404)
(initialized to zeros)

Return: BX:SI buffer filled with nonzero value if installed

Program: COMM-DRV is a universal serial communications driver by Willies'

Computer Software Company, which supports standard INT 14 and
FOSSIL calls as well as its own interfaces

Index: installation check;COMM-DRV

Format of COMM-DRV information area:

Offset Size Description (Table 00404)

00h 8 BYTEs signature "COMM-DRV"

08h 2 BYTEs 00h,00h

0Ah DWORD -> direct address mapping table

0Eh DWORD previous INT 14 vector

-----t-15-----

INT 15 - Microsoft TSR Specification

No additional information available at this time.

-----B-1500-----

INT 15 - CASSETTE - TURN ON TAPE DRIVE'S MOTOR (PC and PCjr only)

AH = 00h

Return: CF set on error

AH = 86h no cassette present

CF clear if successful

SeeAlso: AH=01h"CASSETTE",MEM 0040h:0067h"PC"

-----M-1500-----

INT 15 - Amstrad PC1512 - GET AND RESET MOUSE COUNTS

AH = 00h

Return: CX = signed X count

DX = signed Y count

-----O-1500-----

INT 15 - VMiX v2+ - INSTALLATION CHECK

AH = 00h

Return: DX = 0798h if installed

AX = version (AH = major, AL = minor)

-----T-1500-----

INT 15 - MultiDOS Plus - GIVE UP TIME SLICE

AH = 00h

Return: nothing

Note: if issued by the highest-priority task while MultiDOS is using
priority-based rather than round-robin scheduling, control will be
returned to the caller immediately

SeeAlso: AH=03h"MultiDOS",AX=1000h

-----B-1501-----

INT 15 - CASSETTE - TURN OFF TAPE DRIVE'S MOTOR (PC and PCjr only)

AH = 01h

Return: CF set on error

AH = 86h no cassette present

CF clear if successful

SeeAlso: AH=00h"CASSETTE"

-----b-1501-----

INT 15 - Amstrad PC1512 - WRITE DATA TO NON-VOLATILE RAM

AH = 01h

AL = NVRAM location (00h to 3Fh) (see #00405)

BL = NVRAM data value

Return: AH = return code

00h OK

01h address bad

02h write error

SeeAlso: AH=02h"Amstrad"

Format of Amstrad NVRAM:

Offset Size Description (Table 00405)

00h	BYTE	time of day: seconds
01h	BYTE	alarm time: seconds
02h	BYTE	time of day: minutes
03h	BYTE	alarm time: minutes
04h	BYTE	time of day: hours
05h	BYTE	alarm time: hours
06h	BYTE	day of week, 1 = Sunday
07h	BYTE	day of month
08h	BYTE	month
09h	BYTE	year mod 100
0Ah	BYTE	RTC status register A (see #00406)
0Bh	BYTE	RTC status register B (see #00407)
0Ch	BYTE	RTC status register C (read-only) (see #00408)
0Dh	BYTE	RTC status register D
		bit 7: battery good
0Eh	6 BYTES	time and date machine last used
14h	BYTE	user RAM checksum
15h	WORD	Enter key scancode/ASCII code
17h	WORD	Forward delete key scancode/ASCII code
19h	WORD	Joystick fire button 1 scancode/ASCII code
1Bh	WORD	Joystick fire button 2 scancode/ASCII code
1Dh	WORD	mouse button 1 scancode/ASCII code
1Fh	WORD	mouse button 2 scancode/ASCII code
21h	BYTE	mouse X scaling factor
22h	BYTE	mouse Y scaling factor
23h	BYTE	initial VDU mode and drive count
24h	BYTE	initial VDU character attribute

25h BYTE size of RAM disk in 2K blocks
26h BYTE initial system UART setup byte
27h BYTE initial external UART setup byte
28h 24 BYTES available for user application

Note: bytes 00h-0Dh are the same on the IBM AT as they are used/updated by
the clock chip

Bitfields for RTC status register A:

Bit(s) Description (Table 00406)

7 set if date/time being updated
6-4 time base speed, default 010 = 32768 Hz
3-0 interrupt rate selection, default 0110 = 1024 Hz

SeeAlso: #00405

Bitfields for RTC status register B:

Bit(s) Description (Table 00407)

7 clear if normal update, set if abort update
6 periodic interrupt enable
5 alarm interrupt enable
4 update end interrupt enable
3 square wave enable
2 date mode (clear = BCD, set = binary)
1 24-hour format
0 daylight saving time enable

SeeAlso: #00405

Bitfields for RTC status register C:

Bit(s) Description (Table 00408)

7 IRQF flag
6 PF flag
5 AF flag
4 UF flag

SeeAlso: #00405

-----O-1501-----

INT 15 - VMiX - "sys_chanreq" - I/O CHANNEL OBJECT MANAGER

AH = 01h

STACK: WORD object ID of requestor

DWORD pointer to ASCIZ name of requested method

"assign" assign channel to object

"deassign" deassign channel

"cursor" set cursor on/off

```

"init" initialize comm port
"open" open I/O channel
"position" set cursor position
"receive" get buffered packet from comm port
"send" send buffered packet to comm port
"vio" set current virtual I/O to specified channel
>window" make window at cursor position
---if "assign"---
WORD object UID
WORD caller UID/PID
DWORD CSL with port
---if "deassign"---
WORD channel ID
---if "cursor"---
WORD channel ID (must be a SRCSINK)
WORD new state (0000h off, 0001h on)
---if "init"---
WORD channel ID (must be a SRCSINK)
WORD comm port number (00h-03h)
WORD UART init code
---if "open"---
WORD channel ID
---if "position"---
WORD channel ID (must be a SRCSINK)
WORD position (high byte = row, low byte = column)
---if "receive"---
DWORD pointer to buffer
---if "send"---
WORD length of buffer
DWORD pointer to buffer
---if "vio"---
WORD channel ID (must be a SRCSINK)
---if "window"---
WORD top left (high byte = row, low byte = column)
WORD bottom right (high byte = row, low byte = column)

```

Return: DX:AX -> IRP structure or 0000h:0000h

SeeAlso: AH=00h"VMiX",AH=02h"VMiX"

-----T-1501-----

INT 15 - MultiDOS Plus - REQUEST RESOURCE SEMAPHORE

AH = 01h

AL = semaphore number (00h-3Fh)

Return: AH = status

00h successful

02h invalid semaphore number

Notes: if the semaphore is not owned, ownership is assigned to the calling task and the call returns immediately

if the semaphore is already owned by another task, the calling task is placed on a queue for the semaphore and suspended until it can become owner of the semaphore

semaphore 0 is used internally by MultiDOS to synchronize DOS access

SeeAlso: AH=02h"MultiDOS",AH=10h"MultiDOS",AH=1Bh"MultiDOS"

-----B-1502-----

INT 15 - CASSETTE - READ DATA (PC and PCjr only)

AH = 02h

CX = number of bytes to read

ES:BX -> buffer

Return: CF clear if successful

DX = number of bytes read

ES:BX -> byte following last byte read

CF set on error

AH = status (see #00409)

SeeAlso: AH=00h"CASSETTE",AH=03h"CASSETTE",MEM 0040h:0069h, MEM 0040h:006Bh"PC"

(Table 00409)

Values for Cassette status:

00h successful

01h CRC error

02h bad tape signals

04h no data

80h invalid command

86h no cassette present

-----b-1502-----

INT 15 - Amstrad PC1512 - READ DATA FROM NON-VOLATILE RAM

AH = 02h

AL = NVRAM location (00h to 3Fh)

Return: AH = return code

00h OK

01h address bad

02h checksum error

AL = NVRAM data value

SeeAlso: AH=01h"Amstrad"

-----O-1502-----

INT 15 - VMiX - "sys_memreq" - MEMORY OBJECT MANAGER

AH = 02h

STACK: WORD object ID of requestor

DWORD pointer to ASCIZ name of requested method

"assign" allocate low memory block

"assign extended" allocate extended memory pages

"assign gdt" allocate GDT selector

"paged" allocate low paged memory

"paged extended" allocate extended memory pages

"deassign" free memory block

"deassign gdt" free GDT selector

"getvpage" get physical address for virtual page

"setvpage" set physical address for virtual page

"info" get VMiX memory usage info block

"move" move contents of 32-bit memory

"newmcb" make new DOS memory control block

"owner" get process ID of MCB or PSP owner

"umb" allocate upper memory block

"video" toggle system use of video memory and get stat

---if "assign"---

WORD number of objects

WORD size in bytes (multiple of 512 bytes)

---if "assign extended"---

WORD number of objects

WORD size in bytes (multiple of 4K)

---if "assign gdt"---

WORD access type (low byte)

WORD segment size in paragraphs

DWORD pointer to start of physical segment

---if "paged"---

WORD number of 512-byte pages

---if "paged extended"

WORD number of 4K pages

---if "deassign"---

DWORD pointer returned by previous allocation call

---if "deassign gdt"---

WORD GDT selector

---if "getvpage"---

WORD owner's process ID

DWORD pointer to buffer for page structure (struct VPGE)

---if "setvpage"---

```

WORD owner's process ID
DWORD pointer to new page structure (struct VPGE)
---if "info"---
no additional arguments
---if "move"
DWORD 32-bit source address
DWORD 32-bit destination address
WORD number of words to move
---if "newmcb"---
DWORD pointer to new MCB's location
WORD size of memory block
DWORD pointer to ASCIZ name string (max 8 chars)
---if "owner"---
WORD MCB or PSP segment
---if "umb"---
WORD size in paragraphs
---if "video"---
no additional arguments

```

Return: DX:AX -> memory block or VPGE struct or 0000h:0000h

SeeAlso: AH=00h"VMiX",AH=01h"VMiX"

-----T-1502-----

INT 15 - MultiDOS Plus - RELEASE RESOURCE SEMAPHORE

AH = 02h

AL = semaphore number (00h-3Fh)

Return: AH = status

00h successful

01h not semaphore owner

02h invalid semaphore number

Notes: if any tasks are waiting for the semaphore, the first task on the wait queue will become the new owner and be reawakened

do not use within an interrupt handler

SeeAlso: AH=01h"MultiDOS",AH=10h"MultiDOS",AH=1Ch"MultiDOS"

-----B-1503-----

INT 15 - CASSETTE - WRITE DATA (PC and PCjr only)

AH = 03h

CX = number of bytes to write

ES:BX -> data buffer

Return: CF clear if successful

ES:BX -> byte following last byte written

CF set on error

AH = status (see #00409)

```

CX = 0000h
SeeAlso: AH=00h"CASSETTE",AH=02h"CASSETTE"
-----V-1503-----
INT 15 - Amstrad PC1512 - WRITE VDU COLOR PLANE WRITE REGISTER
  AH = 03h
  AL = value (I,R,G,B bits)
Return: nothing
SeeAlso: AH=04h"Amstrad"
-----O-1503-----
INT 15 - VMiX - "sys_pinput" - PROMPTED CONSOLE INPUT
  AH = 03h
  STACK:  DWORD pointer to ASCII prompt
          WORD  field outline character
          WORD  length of input field (max 7Fh)
          DWORD address of pointer to input buffer
Return: AX = length of input (input buffer is padded with blanks)
SeeAlso: AH=04h"VMiX"
-----T-1503-----
INT 15 - MultiDOS Plus - SUSPEND TASK FOR INTERVAL
  AH = 03h
  DX = number of time slices to remain suspended
Return: after specified interval has elapsed
Note: when priority-based scheduling is in use, high-priority tasks should
      use this function to yield the processor
SeeAlso: AH=00h"MultiDOS",AH=0Ah"MultiDOS"
-----B-1504-----
INT 15 - SYSTEM - BUILD BIOS SYSTEM PARAMETER TABLE (PS)
  AH = 04h
  ES:DI -> 32-byte results buffer for System Parameter Table (see #00410)
  DS = segment containing BIOS RAM extensions (zero if none)
Return: CF clear if successful
       AH = 00h success
       ES:DI buffer filled
       AL destroyed
       CF set on failure
       AX destroyed
       AH = 80h/86h if not supported
SeeAlso: AH=05h"BIOS",AH=C1h

```

```

Format of BIOS System Parameter Table:
Offset  Size  Description (Table 00410)

```

00h DWORD FAR address of BIOS Common Start Routine
04h DWORD FAR address of BIOS Interrupt Routine
08h DWORD FAR address of BIOS Time-out Routine
0Ch WORD number of bytes of stack required by this BIOS implementation
0Eh 16 BYTES reserved
1Eh WORD number of entries in initialization table

-----V-1504-----

INT 15 - Amstrad PC1512 - WRITE VDU COLOR PLANE READ REGISTER

AH = 04h

AL = value (RDSEL1 and RDSEL0)

Return: nothing

SeeAlso: AH=03h"Amstrad",AH=05h"Amstrad"

-----O-1504-----

INT 15 - VMiX - "sys_vprintf" - FORMATTED OUTPUT TO STREAM

AH = 04h

STACK: DWORD control string

DWORD array of arguments

Return: nothing

SeeAlso: AH=03h"VMiX"

-----T-1504-----

INT 15 - MultiDOS Plus - SEND MESSAGE TO ANOTHER TASK

AH = 04h

AL = mailbox number (00h-3Fh)

CX = message length in bytes

DS:SI -> message

Return: AH = status

00h successful

01h out of message memory

02h invalid mailbox number

Note: the message is copied into a system buffer; the caller may immediately reuse its buffer

SeeAlso: AH=05h"MultiDOS"

-----B-1505-----

INT 15 - SYSTEM - BUILD BIOS INITIALIZATION TABLE (PS)

AH = 05h

ES:DI -> results buffer of length 18h * Number_of_Entries (see #00411)

DS = segment containing BIOS RAM extensions (zero if none)

Return: CF clear if successful

AH = 00h success

ES:DI buffer filled

AL destroyed

CF set on failure
 AX destroyed
 AH = 80h/86h if not supported

SeeAlso: AH=04h"ABIOS",AH=C1h

Format of one entry of BIOS Initialization Table:

Offset	Size	Description (Table 00411)
00h	WORD	device ID (see #00412)
02h	WORD	number of Logical IDs
04h	WORD	Device Block length (zero for BIOS patch or extension)
06h	DWORD	-> init routine for Device Block and Function Transfer Table
0Ah	WORD	request block length
0Ch	WORD	Function Transfer Table length (zero for a patch)
0Eh	WORD	Data Pointers length (in Common Data Area)
10h	BYTE	secondary device ID (hardware level this BIOS ver supports)
11h	BYTE	revision (device driver revision level this BIOS supports)
12h	6 BYTES	reserved

(Table 00412)

Values for BIOS device ID:

00h	BIOS internal calls
01h	floppy disk
02h	hard disk
03h	video
04h	keyboard
05h	parallel port
06h	serial port
07h	system timer
08h	real-time clock
09h	system services
0Ah	NMI
0Bh	mouse
0Eh	CMOS RAM
0Fh	DMA
10h	Programmable Option Select (POS)
16h	keyboard password

-----V-1505-----

INT 15 - Amstrad PC1512 - WRITE VDU GRAPHICS BORDER REGISTER

AH = 05h

AL = value (I,R,G,B bits)

Return: nothing

SeeAlso: AH=04h"Amstrad"
-----O-1505-----
INT 15 - VMiX - "sys_getpid" - GET PROCESS ID OF CURRENT PROCESS
AH = 05h
Return: AX = process ID
SeeAlso: AH=06h"VMiX",AH=0Bh"VMiX"
-----T-1505-----
INT 15 - MultiDOS Plus - CHECK MAILBOX
AH = 05h
AL = mailbox number (00h-3Fh)
Return: AH = status
00h successful
DX = length of first message in queue, 0000h if no message
02h invalid mailbox number
SeeAlso: AH=04h"MultiDOS",AH=06h"MultiDOS"
-----b-1506-----
INT 15 - Amstrad PC1512 - GET ROS VERSION NUMBER
AH = 06h
Return: BX = version number
-----O-1506-----
INT 15 - VMiX - "sys_getpcb" - GET POINTER TO PROCESS CONTROL BLOCK
AH = 06h
STACK: WORD process ID
Return: DX:AX -> process control block
SeeAlso: AH=05h"VMiX",AH=07h"VMiX",AH=08h"VMiX"
-----T-1506-----
INT 15 - MultiDOS Plus - READ MAILBOX
AH = 06h
AL = mailbox number (00h-3Fh)
CX = size of buffer in bytes
ES:DI -> buffer for message
Return: AH = status
00h successful
CX = number of bytes copied
DX = actual length of message
02h invalid mailbox number
Note: if the caller's buffer is not large enough, the message is truncated
and the remainder is lost
SeeAlso: AH=04h"MultiDOS",AH=05h"MultiDOS"
-----O-1507-----
INT 15 - VMiX - "sys_getocb" - GET POINTER TO OBJECT CONTROL BLOCK

```
AH = 07h
STACK: WORD object type
Return: DX:AX -> object control block
SeeAlso: AH=06h"VMiX",AH=08h"VMiX"
-----1507-----
INT 15 - IBM SurePath BIOS - Officially "Private" Function
AH = 07h
SeeAlso: AH=08h"IBM"
-----T-1507-----
INT 15 - MultiDOS Plus - SPAWN INTERNAL TASK (CREATE NEW THREAD)
AH = 07h
BX:CX = entry point of new task
DX = stack size in paragraphs
Return: AH = status
    00h successful
    01h no free task control blocks
    02h no free memory for task's stack
Note: execution returns immediately to calling task
SeeAlso: AH=08h"MultiDOS",AH=09h"MultiDOS",AH=13h"MultiDOS"
-----O-1508-----
INT 15 - VMiX - "sys_getccb" - GET CHANNEL CONTROL BLOCK
AH = 08h
STACK: WORD channel ID
Return: DX:AX -> channel control block
SeeAlso: AH=06h"VMiX",AH=07h"VMiX"
-----B-1508-----
INT 15 - IBM SurePath BIOS - WAIT REQUESTED TIME PERIOD
AH = 08h
AL = function
    00h wait in increments of 15.025 microseconds
CX = number of time increments to wait (0000h = maximum)
    80h wait in increments of 840 ns
ECX = number of time increments to wait
    81h I/O event wait
BH = bitmask of bits to check
BL = expected pattern
DX = I/O port address
ECX = number of 840 ns microticks to wait
Return: ECX = 00000000h if expected pattern did not occur
    82h memory event wait
BH = bitmask of bits to check
```

BL = expected pattern
ES:SI -> BYTE to check
ECX = number of 840 ns microticks to wait
Return: ECX = 00000000h if expected pattern did not occur
other reserved

Return: CF clear if successful

CF set on error

AH = status

00h successful

01h used 15.025 microsecond interval, time rounded up

08h reserved subfunction

86h function not supported

Notes: IBM classifies this function as optional

if the POST determines that the timer is nonfunctional, this function
uses the 15.025 microsecond refresh timer instead of the
full-resolution timer

SeeAlso: AH=07h"IBM",AH=09h"IBM",AH=86h

-----T-1508-----

INT 15 - MultiDOS Plus - TERMINATE INTERNAL TASK (KILL THREAD)

AH = 08h

Return: calling task terminated, so execution never returns to caller

Notes: an internal task must be terminated with this function rather than a
DOS termination function

task's stack space is returned to parent task's memory pool

SeeAlso: AH=07h"MultiDOS"

-----O-1509-----

INT 15 - VMiX - "sys_getqueue" - GET ID OF QUEUED ELEMENT

AH = 09h

STACK: WORD queue ID (0 = process queue, 1 = object, 3 = type)

WORD subqueue ID

Return: AX = queue ID

SeeAlso: AH=0Ah"VMiX"

-----1509-----

INT 15 - IBM BIOS - RESERVED FOR PCMCIA SYSTEM RESOURCE TABLE ACCESS

AH = 09h

no further details available

SeeAlso: AH=08h"IBM"

-----T-1509-----

INT 15 - MultiDOS Plus - CHANGE TASK'S PRIORITY

AH = 09h

AL = new priority

Return: nothing

Note: the priority has different meanings depending on whether priority-based or round-robin scheduling is used

SeeAlso: AH=07h"MultiDOS"

-----O-150A-----

INT 15 - VMiX - "sys_getqnext" - GET ID OF NEXT QUEUED ELEMENT

AH = 0Ah

STACK: WORD queue ID (0 = process queue, 1 = object, 3 = type)

WORD ID of current element in queue chain

Return: AX = ID of next element

SeeAlso: AH=09h"VMiX",AH=0Fh"VMiX"

-----T-150A-----

INT 15 - MultiDOS Plus - CHANGE TIME SLICE INTERVAL

AH = 0Ah

AL = new interval

00h = 55.0 ms (default)

80h = 27.5 ms

40h = 13.75 ms

20h = 6.88 ms

10h = 3.44 ms

08h = 1.72 ms

SeeAlso: AH=03h"MultiDOS"

-----O-150B-----

INT 15 - VMiX - "sys_sysreq" - SYSTEM CONFIGURATION MANAGER

AH = 0Bh

STACK: WORD caller's UID

DWORD pointer to ASCIZ name of requested method

"abort" abort current send/receive on comm port

"block" start/end critical section

"close" terminate interrupt-drive comm I/O

"open" prepare comm port for interrupt-driven I/O

"delay" set delay timer and wait

"hibernate" put process to sleep

"ints" enable/disable interrupt-driven INT 14h

"length" get current send/receive buffer offsets

"kswitch" switch stacks

"numproc" get number of active processes

"protocol" set protocol function for comm interrupts

"relocate" set/reset VMiX flag for relocating to himem

"status" get current open comm port status

"wake" awaken a process

```

"xport" get comm port polled for logins
---if "abort"---
no additional arguments
---if "block"---
WORD 0000h end, 0001h start
---if "close"---
no additional arguments
---if "open"---
WORD comm port (00h-03h)
WORD BIOS parameter byte (see #00300 at INT 14/AH=00h),
  except bits 7-5: 000 = 19200, 001 = 38400, 011 = 115200
---if "delay"---
WORD time in seconds
---if "hibernate"---
WORD process ID
---if "ints"---
WORD 0000h if no, 0001h if yes
---if "length","numproc","relocate","status","xport"---
no additional arguments
---if "kswitch"---
DWORD pointer to new stack
---if "protocol"---
DWORD pointer to function (must be in low "assign"ed memory
  when in 386 mode)
---if "wake"---
WORD process ID

```

Return: DX:AX -> result or 0000h:0000h

```

---if "length"---
BYTE receive offset
BYTE send offset
---if "kswitch"---
DWORD old stack pointer
---if "numproc"---
WORD number of active processes
---if "status"---
current open comm port status
---if "xport"---
current comm port being polled for logins

```

Note: the "delay" command reportedly disables the keyboard until the delay completes

SeeAlso: AH=05h"VMiX",AH=0Eh"VMiX"

-----T-150B-----

INT 15 - MultiDOS Plus - FORCE DISPLAY OUTPUT TO PHYSICAL SCREEN MEMORY

AH = 0Bh

Return: nothing

Notes: sets calling task's screen pointer to actual screen memory; the pointer
may be restored with AH=0Ch
caller's video mode must be same as foreground task's video mode
any text written while in the background will be saved to the
foreground task's virtual screen when it switches to the background
useful if a background task wants to display a message on the
foreground screen

SeeAlso: AH=0Ch"MultiDOS"

-----O-150C-----

INT 15 - VMiX - "sys_getstack" - GET POINTER TO PROCESS TSS STACK

AH = 0Ch

STACK: WORD process ID

Return: DX:AX -> TSS stack store

SeeAlso: AH=00h"VMiX"

-----T-150C-----

INT 15 - MultiDOS Plus - RESTORE OLD VIDEO DISPLAY MEMORY

AH = 0Ch

Return: nothing

Note: restores task's screen pointer saved by AH=0Bh; must not be called
unless AH=0Bh has been called first

SeeAlso: AH=0Bh"MultiDOS"

-----O-150D-----

INT 15 - VMiX - "sys_spawn" - START A CHILD PROCESS JOB SHELL

AH = 0Dh

STACK: DWORD ASCIZ string starting with requested I/O channel and
followed by standard VMiX shell command string

Return: AX = process ID or error code "SYS_ERROR"

Note: the maximum string length is 7Fh characters

SeeAlso: AH=0Eh"VMiX",AH=11h"VMiX",INT 21/AH=4Bh

-----T-150D-----

INT 15 - MultiDOS Plus - DISABLE MULTITASKING

AH = 0Dh

Return: nothing

Note: calling task receives all time slices until AH=0Eh is called; this
allows time-critical events or nonreentrant code to be processed

SeeAlso: AH=0Eh"MultiDOS",AH=10h"MultiDOS",AX=101Bh,AH=20h"MultiDOS"

-----O-150E-----

INT 15 - VMiX - "sys_kill" - HARD TERMINATE PROCESS

AH = 0Eh

STACK: WORD process ID

Return: AX = status (SYS_OK or SYS_ERROR)

SeeAlso: AH=0Bh"VMiX",AH=0Dh"VMIX"

-----T-150E-----

INT 15 - MultiDOS Plus - ENABLE MULTITASKING

AH = 0Eh

Return: nothing

SeeAlso: AH=0Dh"MultiDOS",AX=101Ch,AH=20h"MultiDOS"

-----d-150F-----

INT 15 C - SYSTEM - FORMAT UNIT PERIODIC INTERRUPT (PS ESDI drives only)

AH = 0Fh

AL = phase code

00h reserved

01h surface analysis

02h formatting

Return: CF clear if formatting should continue

CF set if formatting should terminate

Note: called during ESDI drive formatting after each cylinder is completed

SeeAlso: INT 13/AH=1Ah

-----O-150F-----

INT 15 - VMiX - "sys_getqkey" - GET KEY FIELD OF QUEUED ELEMENT

AH = 0Fh

STACK: WORD queue ID (0 = process queue, 1 = object q, 3 = type q)

WORD ID of element in queue chain

Return: AX = key

SeeAlso: AH=0Ah"VMiX"

-----T-150F-----

INT 15 - MultiDOS Plus - EXECUTE A MULTIDOS PLUS COMMAND

AH = 0Fh

DS:BX -> ASCIZ command

Return: after command has been processed

Notes: specified string is executed as if it had been typed at the MultiDOS
command prompt

the task is placed on a queue which MultiDOS examines periodically and
is suspended until MultiDOS has processed the command

all lowercase characters up to the first blank are converted to upper
case within the given buffer

-----!---Section-----