

Interrupt List Release 61 Last change 16jul00

Copyright (c) 1989-1999,2000 Ralf Brown

-----!---FILELIST-----

Please redistribute the following files unmodified as a group, in a quartet of archives named INTER61A through INTER61D (preferably the original authenticated PKZIP archives):

INTERRUPT.1ST the read-me file, containing credits, availability info
INTERRUPT.A INT 00 through INT 10/BE \
INTERRUPT.B INT 10/BF through INT 15/0F\
INTERRUPT.C INT 15/10 through INT 15/E7 \
INTERRUPT.D INT 15/E8 through INT 1A/B0 \
INTERRUPT.E INT 1A/B1 through INT 1F \
INTERRUPT.F INT 20 through INT 21/43 \
INTERRUPT.G INT 21/44 through INT 21/5E \
INTERRUPT.H INT 21/5F through INT 21/E2 >(8754 entries)
INTERRUPT.I INT 21/E3 through INT 21/F1 / (4135 tables)
INTERRUPT.J INT 21/F2 through INT 25 /
INTERRUPT.K INT 26 through INT 2F/15 /
INTERRUPT.L INT 2F/16 through INT 2F/79 /
INTERRUPT.M INT 2F/7A through INT 2F/D9 /
INTERRUPT.N INT 2F/DA through INT 50 /
INTERRUPT.O INT 51 through INT 61 /
INTERRUPT.P INT 62 through INT 6A /
INTERRUPT.Q INT 6B through INT 91 /
INTERRUPT.R INT 92 through INT FF /
INTERRUPT.PRI a brief introduction to interrupts
INTPRINT.COM a simple formatter that also generates a list summary
INTPRINT.DOC instructions for INTPRINT
OVERVIEW.LST brief listing of major uses of each interrupt
86BUGS.LST a listing of CPU bugs and undocumented features
BIBLIO.LST bibliography of information sources for the list
CMOS.LST a description of the CMOS RAM data bytes
FARCALL.LST APIs available through FAR CALLs
GLOSSARY.LST a glossary of terms, abbreviations, and acronyms
MEMORY.LST format of the BIOS data area
OPCODES.LST a listing of recent and undocumented CPU instructions
PORTS.A a listing of I/O ports 0000h-013Fh
PORTS.B a listing of I/O ports 0140h-0AD5h
PORTS.C a listing of I/O ports 0AD6h-FFFFh
CATEGORY.KEY descriptions of divider-line category letters
COMBINE.COM combine the pieces of the list into a single file

COMBINE.DOC documentation for COMBINE

The following files should be distributed in an archive called INTER61E:

86BUGSnn.ZIP programs to test for CPU bugs
COMBINE.ASM source code for COMBINE.COM
INT.* invoke interrupts from commandline
INTHLP???.ZIP Interrupt Helper viewer for the interrupt list
INTLIST.E Epsilon extension for handling list
INTPRINT.C source code for INTPRINT
INTSUM???.ZIP interrupt list browser (Interrupt Summary)
IVIEW*.ZIP another interrupt list browser

Finally, the following should be distributed in an archive called INTER61F:

INT2RTF.ZIP Slava Gostrenko's Windows Help converter
INT2WHLP.ZIP convert list into Windows Help database
INTERRUP.ICO icon for interrupt list
HINTSRCH.ZIP WinHelp DLL for full-text searches of interrupt list
IL2ME???.ZIP convert list into Multi-Edit help database
INT2GUID.* convert list into TurboPower GUIDE or POPHELP database
INT2HLP.ZIP convert list into QuickHelp database
INT2IPF.ZIP convert list into OS/2 .IPF database
INT2QH.* program to convert list into QuickHelp database
INT2TPH.ZIP convert to Turbo/Borland Pascal help file (.TPH)
INTHELP.* convert list into TurboPower GUIDE database
RB2NG???.ZIP convert list into Norton Guides database
WH_ED*.ZIP WinHelp-file editor

-----!---CONTACT_INFO-----

If you notice any mistakes or omissions, please let me know! It is only with YOUR help that the list can continue to grow at the current rate. Please send all changes to me rather than distributing a modified version of the list.

Please read the file INTERRUP.1ST before asking me any questions. You may find that they have already been addressed.

Ralf Brown

Internet: ralf@pobox.com (currently forwards to ralf@telerama.lm.com)

FIDO: Ralf Brown 1:129/26.1

or post a message to me in the DR_DEBUG echo (I probably won't see it unless you address it to me)

I reply to all e-mail submissions, but some of my replies bounce because of bad return paths. If you don't get a response from me within a reasonable

period of time, send it again with a better return path (many mailers are improperly configured and do not produce a valid From: address).

-----!---Note-----

See INTERRUPT.1ST for the key to system abbreviations and a list of the trademarks mentioned here.

-----!---DISCLAIMER-----

DISCLAIMER: THIS MATERIAL IS PROVIDED "AS IS". I verify the information contained in this list to the best of my ability, but I cannot be held responsible for any problems caused by use or misuse of the information, especially for those functions not officially documented. If it is marked "internal" or undocumented, you should check it carefully to make sure it works the same way in your version of the software (and please let me know whether or not it works the same way). Information marked with "???" is known to be incomplete or guesswork.

-----!---FLAGS-----

The use of -> instead of = signifies that the indicated register or register pair contains a pointer to the specified item, rather than the item itself. Register pairs (such as AX:BX) indicate that the item is split across the registers, with the high-order half in the first register.

One or more letters may follow the interrupt number; they have the following meanings: U - undocumented function, u - partially documented function, P - available only in protected mode, R - available only in real or V86 mode, C - callout or callback (usually hooked rather than called), O - obsolete (no longer present in current versions)

-----!---CATEGORIES-----

The ninth column of the divider line preceding an entry usually contains a classification code (the entry has not been classified if that character is a dash). The codes currently in use are:

- A - applications, a - access software (screen readers, etc),
- B - BIOS, b - vendor-specific BIOS extensions,
- C - CPU-generated, c - caches/spoolers,
- D - DOS kernel, d - disk I/O enhancements,
- E - DOS extenders, e - electronic mail, F - FAX,
- f - file manipulation, G - debuggers/debugging tools, g - games,
- H - hardware, h - vendor-specific hardware,
- I - IBM workstation/terminal emulators, i - system info/monitoring,
- J - Japanese, j - joke programs,
- K - keyboard enhancers, k - file/disk compression,
- l - shells/command interpreters,
- M - mouse/pointing device, m - memory management,

N - network, n - non-traditional input devices,
O - other operating systems,
P - printer enhancements, p - power management,
Q - DESQview/TopView and Quarterdeck programs,
R - remote control/file access, r - runtime support,
S - serial I/O, s - sound/speech,
T - DOS-based task switchers/multitaskers, t - TSR libraries
U - resident utilities, u - emulators,
V - video, v - virus/antivirus,
W - MS Windows,
X - expansion bus BIOSes, x - non-volatile config storage
y - security, * - reserved (and not otherwise classified)

-----C-00-----

INT 00 C - CPU-generated - DIVIDE ERROR

Desc: generated if the divisor of a DIV or IDIV instruction is zero or the quotient overflows the result register; DX and AX will be unchanged.

Notes: on an 8086/8088, the return address points to the following instruction
on an 80286+, the return address points to the divide instruction
an 8086/8088 will generate this interrupt if the result of a division is 80h (byte) or 8000h (word)

SeeAlso: INT 04,OPCODE "AAD"

-----G-00-----

INT 00 - Zenith - ROM DEBUGGER

Desc: invokes the ROM Debugger when at the BIOS level; equivalent to pressing Ctrl-Alt-Ins on booting.

Note: since DOS revector INT 00, it is necessary to restore this vector to its original ROM BIOS value in order to invoke the debugger once DOS loads

SeeAlso: INT 03"Columbia"

-----C-01-----

INT 01 C - CPU-generated - SINGLE STEP

Desc: generated after each instruction if TF (trap flag) is set; TF is cleared on invoking the single-step interrupt handler

Notes: interrupts are prioritized such that external interrupts are invoked after the INT 01 pushes CS:IP/FLAGS and clears TF, but before the first instruction of the handler executes
used by debuggers for single-instruction execution tracing, such as MS-DOS DEBUG's T command

SeeAlso: INT 03"CPU"

-----C-01-----

INT 01 C - CPU-generated (80386+) - DEBUGGING EXCEPTIONS

Desc: generated by the CPU on various occurrences which may be of interest
to a debugger program

Note: events which may trigger the interrupt:

Instruction address breakpoint fault - will return to execute inst
Data address breakpoint trap - will return to following instruction
General detect fault, debug registers in use
Task-switch breakpoint trap
undocumented 386/486 opcode Flh - will return to following instruc

SeeAlso: INT 03"CPU"

-----H-02-----

INT 02 C - external hardware - NON-MASKABLE INTERRUPT

Desc: generated by the CPU when the input to the NMI pin is asserted

Notes: return address points to start of interrupted instruction on 80286+

on the 80286+, further NMIs are disabled until the next IRET
instruction, but one additional NMI is remembered by the hardware
and will be serviced after the IRET instruction reenables NMIs
maskable interrupts may interrupt the NMI handler if interrupts are
enabled

although the Intel documentation states that this interrupt is
typically used for power-failure procedures, it has many other uses
on IBM-compatible machines:

Memory parity error: all except Jr, CONV, and some machines
without memory parity

Breakout switch on hardware debuggers

Coprocessor interrupt: all except Jr and CONV

Keyboard interrupt: Jr, CONV

I/O channel check: CONV, PS50+

Disk-controller power-on request: CONV

System suspend: CONV

Real-time clock: CONV

System watch-dog timer, time-out interrupt: PS50+

DMA timer time-out interrupt: PS50+

Low battery: HP 95LX

Module pulled: HP 95LX

-----m-02----SI0714-----

INT 02 U - STB RAPIDMAP.SYS - ???

SI = 0714h

ES:DI -> ???

Return: ???

SeeAlso: INT 67/AX=6100h"STB",PORT 00E1h"STB"

-----C-03-----

INT 03 C - CPU-generated - BREAKPOINT

Desc: generated by the one-byte breakpoint instruction (opcode CCh)

Notes: used by debuggers to implement breakpoints, such as MS-DOS DEBUG's G command

also used by Turbo Pascal versions 1,2,3 when {\$U+} specified
return address points to byte following the breakpoint instruction
called by Novell DOS 7 EMM386 after displaying an exception error
message (GPF, illegal opcodes, etc.) and before waiting for user
input

called by QEMM-386 if the user presses 'I' (undocumented) when prompted
after an exception error message

SeeAlso: INT 01"CPU"

-----G-03-----

INT 03 - Columbia PCs (desktop,VP portables) - ROM DEBUGGER

Desc: invokes the ROM Debugger if INT 03 has not been revectorized; equivalent
to pressing Esc on booting.

SeeAlso: INT 00"Zenith",INT 03"Realia"

-----G-03-----

INT 03 - Realia COBOL - DEBUGGER SUPPORT

Note: Realia COBOL checks for a signature eight bytes prior to the interrupt
handler when it starts up

BUG: if the offset of the INT 03 vector is less than 8, the compiler and
all executables generated by it will crash the system

SeeAlso: INT 03"Columbia"

-----G-03-----

INT 03 U - Watcom WVIDEO, Watcom WD - OUTPUT DEBUGGING MESSAGE

STACK: DWORD selector

DWORD offset of ASCIZ message to display

Notes: the INT 03h instruction which invokes the debugger must be followed by
a JMP SHORT and the signature string "WVIDEO"

to check whether this interface is available, define a public byte
variable named __WD_Present, which the debugger will set to 01h
before running the program

SeeAlso: AX=0910h

-----U-03-----

INT 03 - DTown Utilities - POP UP

Program: DTown Software Development's DTown Utilities contains various tools
which are useful to programmers, such as a memory viewer and
disassembler

Note: DTU pops up on INT 03 (if not hooked by a debugger) to allow the
user to inspect memory or code when the machine crashes; various

memory managers will invoke INT 03 when displaying a CPU exception error message (either automatically or when the user presses a particular key)

SeeAlso: INT 2D/AL=20h"DTown"

-----G-03000-----

INT 03 - Soft-ICE - BACK DOOR COMMANDS - GET Soft-ICE VERSION

AX = 0000h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

Return: SI = BCD version (0280h = v2.80)

SeeAlso: INT 03/AX=0910h

-----G-0301-----

INT 03 - Soft-ICE - BACK DOOR COMMANDS - ???

AH = 01h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

Return: ???

SeeAlso: AX=0000h,AX=090Bh,INT 03/AH=10h

-----G-030900-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 0900h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

DS:BX -> ???

CX = ???

DX = ???

Return: ???

SeeAlso: AX=0000h,AH=01h,AX=0902h,AX=0914h

-----G-030902-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - POPUP & START A DEBUG SESSION

AX = 0902h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

DS:BX -> initial register values (see #00001)

Return: registers as specified in initial register values table

Note: This function is called by LDR.EXE in loading a program to be debugged. After executing the function, Soft-ICE pops up its debugging window and you may start debugging your program.

Return: ???

SeeAlso: AX=0000h,AX=090Ah,AX=0910h,AX=0914h

Format of Soft-ICE initial register values:

Offset Size Description (Table 00001)

00h	WORD	initial value of SP
02h	WORD	initial value of SS
04h	WORD	initial value of IP
06h	WORD	initial value of CS
08h	WORD	initial value of DS and ES
0Ah	WORD	???
0Ch	WORD	???
0Eh	WORD	???
10h	WORD	initial value of AX
12h	WORD	??? (defaults to 000AH ???)
14h	WORD	??? (defaults to 0001H ???)
16h	WORD	??? (defaults to 0100H ???)

-----G-030903-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 0903h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

DS:BX -> ???

CX = ???

DX = ???

Return: ???

SeeAlso: AX=0000h,AX=0902h,AX=0907h,AX=0914h

-----G-030907-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 0907h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

DS:BX -> ???

CX = ???

DX = ???

Return: ???

SeeAlso: AX=0000h,AX=0903h,AX=090Ah,AX=0914h

-----G-03090A-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 090Ah

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

DS:BX -> ???

CX = ???


```
DX = ???
Return: ???
SeeAlso: AX=0000h,AX=0907h,AX=090Bh,AX=0914h
-----G-03090B-----
INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???
  AX = 090Bh
  SI = magic value 4647h ('FG')
  DI = magic value 4A4Dh ('JM')
  DS:BX -> ???
  CX = ???
  DX = ???
Return: ???
SeeAlso: AX=0000h,AX=090Ah,AX=0910h,AX=0914h
-----G-030910-----
INT 03 - Soft-ICE - BACK DOOR COMMANDS - DISPLAY STRING IN Soft-ICE WINDOW
  AX = 0910h
  SI = magic value 4647h ('FG')
  DI = magic value 4A4Dh ('JM')
  DS:DX -> ASCIZ string to display (max 100 bytes, 0Dh OK)
Program: Soft-ICE is a debugger by Nu-Mega Technologies, Inc.
SeeAlso: INT 03"WVIDEO"
-----G-030911-----
INT 03 - Soft-ICE - BACK DOOR COMMANDS - EXECUTE Soft-ICE COMMAND
  AX = 0911h
  SI = magic value 4647h ('FG')
  DI = magic value 4A4Dh ('JM')
  DS:DX -> ASCIZ command string (max 100 bytes, 0Dh OK)
Return: nothing
SeeAlso: AX=0912h,AX=0913h
-----G-030912-----
INT 03 - Soft-ICE - BACK DOOR COMMANDS - GET BREAKPOINT INFORMATION
  AX = 0912h
  SI = magic value 4647h ('FG')
  DI = magic value 4A4Dh ('JM')
Return: BH = entry number of last breakpoint set
      BL = type of last breakpoint set (see #00002)
      DH = entry number of last breakpoint to be triggered
      DL = type of last triggered breakpoint (see #00002)
Program: Soft-ICE is a debugger by Nu-Mega Technologies, Inc.
SeeAlso: AX=0911h,AX=0913h,AX=0914h
```

(Table 00002)

Values for Soft-ICE breakpoint type:

00h BPM (breakpoint register types)
 01h I/O
 02h INTerrupt
 03h BPX (INT 03-style breakpoint)
 04h reserved
 05h range

-----G-030913-----

INT 03 - Soft-ICE v2.5x - BACK DOOR COMMANDS - SET Soft-ICE BREAKPOINT

AX = 0913h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

DS:DX -> breakpoint structure (see #00003)

Return: AX = status

00h successful

BX = breakpoint number

03h breakpoint table full

06h memory limit error

07h I/O limit error

09h range limit error

16h duplicate breakpoint

SeeAlso: AX=0911h,AX=0912h,AX=0914h

Format of Soft-ICE breakpoint structure:

Offset Size Description (Table 00003)

00h BYTE breakpoint type (see #00004)

01h DWORD breakpoint address 1

(lower range limit for memory BPs,
 interrupt number for interrupt BPs,
 address of BP for execution BPs,
 I/O address (only word) for I/O BPs)

05h DWORD breakpoint address 2

(upper range limit for memory BPs,
 optional value to check for for interrupt BPs,
 overlay number (0 = root) for execution BPs)

09h DWORD breakpoint address 3

0Dh BYTE breakpoint mode 1 (see #00005)

(for interrupt BPs = register to check

00h no value checking

01h check AL

```

    02h  check AH
    03h  check AX)
0Eh  BYTE  breakpoint mode 2 (see #00005)
0Fh  BYTE  breakpoint size (00h byte, 01h word, 03h dword)
10h  BYTE  breakpoint pass count before program stop
11h  BYTE  breakpoint state

```

Note: all unused fields should contain zeros

(Table 00004)

Values for Soft-ICE breakpoint type:

```

00h  memory location
01h  memory range
03h  I/O
04h  interrupt
05h  execution break

```

(Table 00005)

Values for Soft-ICE breakpoint mode:

```

01h  read
02h  write
04h  execution

```

-----G-030914-----

INT 03 - Soft-ICE v2.5x - BACK DOOR COMMANDS - REMOVE Soft-ICE BREAKPOINT

```

  AX = 0914h
  SI = magic value 4647h ('FG')
  DI = magic value 4A4Dh ('JM')
  BX = breakpoint number (returned by AX=0913h)

```

Return: BX = ???

Program: Soft-ICE is a debugger by Nu-Mega Technologies, Inc.

SeeAlso: AX=0912h,AX=0913h

-----G-030918-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

```

  AX = 0918h
  SI = magic value 4647h ('FG')
  DI = magic value 4A4Dh ('JM')
  DS:BX -> ???
  CX = ???
  DX = ???

```

Return: ???

SeeAlso: INT 03/AX=0000h,INT 03/AX=0910h,INT 03/AX=0914h,INT 03/AH=10h

-----G-0310-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AH = 10h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

Return: ???

SeeAlso: INT 03/AX=0000h, INT 03/AX=0910h, INT 03/AH=01h, INT 03/AH=11h

-----G-0311-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AH = 11h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

Return: none???

SeeAlso: INT 03/AX=0000h, INT 03/AX=0910h, INT 03/AH=10h, INT 03/AX=130Ch

-----G-03130C-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 130Ch

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

BX = ???

CX = ???

DX = ???

Return: AX = ??? (1300h)

SeeAlso: INT 03/AX=0000h, INT 03/AX=0910h, INT 03/AH=11h, INT 03/AX=130Eh

-----G-03130E-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 130Eh

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

BX = ???

CX = ???

DX = ???

Return: AX = ??? (1300h)

SeeAlso: INT 03/AX=0000h, INT 03/AX=0910h, INT 03/AX=130Ch, INT 03/AX=1313h

-----G-031313-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 1313h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

BX = ???

CX = ???

DX = ???

Return: AX = ??? (1300h)

SeeAlso: INT 03/AX=0000h,INT 03/AX=0910h,INT 03/AX=130Eh

-----G-030900-----

INT 03 - Soft-ICE v2.80 - BACK DOOR COMMANDS - ???

AX = 0900h

SI = magic value 4647h ('FG')

DI = magic value 4A4Dh ('JM')

Return: none???

SeeAlso: INT 03/AX=0000h,INT 03/AX=0910h,INT 03/AH=10h,INT 03/AH=12h

-----C-04-----

INT 04 C - CPU-generated - INTO DETECTED OVERFLOW

Desc: the INTO instruction will generate this interrupt if OF (Overflow Flag)

is set; otherwise, INTO is effectively a NOP

Note: may be used for convenient overflow testing (to prevent errors from propagating) instead of JO or a JNO/JMP combination

SeeAlso: INT 00"CPU",OPCODE "AAD",OPCODE "AAM"

-----B-05-----

INT 05 - PRINT SCREEN

Desc: dump the current text screen to the first printer

Notes: normally invoked by the INT 09 handler when PrtSc key is pressed, but may be invoked directly by applications

byte at 0050h:0000h contains status used by default handler

00h not active

01h PrtSc in progress

FFh last PrtSc encountered error

default handler is at F000h:FF54h in IBM PC and 100%-compatible BIOSes

since the BOUND instruction also calls INT 05h, but returns control to the BOUND instruction, a failed BOUND check will cause an infinite loop of PrtScreens unless the INT 05 handler is aware of the problem and checks whether the interrupt was invoked by a BOUND instruction

BUG: some old BIOSes/applications appear to destroy BP on return

SeeAlso: INT 10/AH=12h/BL=20h,INT 4A"Tandy 2000",INT 5E"TI Professional"

SeeAlso: INT 80"Phar Lap"

-----C-05-----

INT 05 C - CPU-generated (80186+) - BOUND RANGE EXCEEDED

Desc: generated by BOUND instruction when the value to be tested is less than the indicated lower bound or greater than the indicated upper bound.

Note: returning from this interrupt re-executes the failing BOUND instruction

-----P-05454A-----

INT 05 U - PSPS v2.01 - EJECT PAGE

AX = 454Ah ('EJ')

Program: PSPS is shareware PostScript PrintScreen utility by A.N.D.

Technologies

Note: sends a ^D to the current printer

SeeAlso: AX=4E57h,AX=5053h,AX=554Eh

-----P-054E57-----

INT 05 U - PSPS v2.01 - SET PARAMETERS

AX = 4E57h ('NW')

ES:SI -> 11-byte buffer containing new settings

Return: buffer filled

SeeAlso: AX=454Ah,AX=5053h,INT 17/AH=63h

-----P-055053-----

INT 05 U - PSPS v2.01 - GET PARAMETERS

AX = 5053h ('PS')

ES:SI -> 11-byte buffer for settings

Return: buffer filled

SeeAlso: AX=4E57h,AX=554Eh,INT 17/AH=64h

-----P-05554E-----

INT 05 U - PSPS v2.01 - UNINSTALL

AX = 554Eh ('UN')

Program: PSPS is shareware PostScript PrintScreen utility by A.N.D.

Technologies

SeeAlso: AX=454Ah,AX=4E57h,INT 17/AH=62h

-----C-06-----

INT 06 C - CPU-generated (80186+) - INVALID OPCODE

Desc: this interrupt is generated when the CPU attempts to execute an invalid opcode (most protected-mode instructions are considered invalid in real mode) or a BOUND, LDS, LES, or LIDT instruction which specifies a register rather than a memory address

Notes: return address points to beginning of invalid instruction with proper programming, this interrupt may be used to emulate instructions which do not exist; many 386 BIOSes emulate the 80286 undocumented LOADALL instruction which was removed from the 80386+ generated by the 80386+ when the LOCK prefix is used with instructions other than BTS, BTR, BTC, XCHG, XADD (486), CMPXCHG (486), INC, DEC, NOT, NEG, ADD, ADC, SUB, SBB, AND, OR, or XOR, or any instruction not accessing memory.

SeeAlso: INT 0C"CPU",INT 0D"CPU"

-----b-06-----

INT 06 C - HP 95LX - SLEEP/WAKEUP

Desc: this interrupt is called just before going into light or deep (shutdown) sleep and just after returning from light or deep sleep

SeeAlso: INT 0B"HP 95LX",INT 15/AX=4DD4h,INT 15/AH=4Eh

-----V-06-----

INT 06 U - no-name clone - GRAPHICS CHARACTER TABLE (NOT A VECTOR!)

Desc: points at graphics data for characters 00h-7Fh of the current font

SeeAlso: INT 43"EGA",INT 44"VIDEO"

-----C-07-----

INT 07 C - CPU-generated (80286+) - PROCESSOR EXTENSION NOT AVAILABLE

Desc: this interrupt is automatically called if a coprocessor instruction is encountered when no coprocessor is installed

Note: can be used to emulate a numeric coprocessor in software

SeeAlso: INT 09"CPU"

-----H-08-----

INT 08 C - IRQ0 - SYSTEM TIMER

Desc: generated 18.2 times per second by channel 0 of the 8254 system timer, this interrupt is used to keep the time-of-day clock updated

Notes: programs which need to be invoked regularly should use INT 1C unless they need to reprogram the timer while still keeping the time-of-day clock running at the proper rate

default handler is at F000h:FEA5h in IBM PC and 100%-compatible BIOSes

may be masked by setting bit 0 on I/O port 21h

SeeAlso: INT 1C,INT 4A"SYSTEM",INT 50"DESQview",INT 58"DoubleDOS",INT 70"IRQ8"

SeeAlso: INT 78"GO32",INT D8"Screen Thief",MEM 0040h:0040h,MEM 0040h:006Ch

-----C-08-----

INT 08 C - CPU-generated (80286+) - DOUBLE EXCEPTION DETECTED

Desc: called when multiple exceptions occur on one instruction, or an exception occurs in an exception handler

Notes: called in protected mode if an interrupt above the defined limit of the interrupt vector table occurs

return address points at beginning of instruction with errors or the beginning of the instruction which was about to execute when the external interrupt caused the exception

if an exception occurs in the double fault handler, the CPU goes into SHUTDOWN mode (which circuitry in the PC/AT converts to a reset); this "triple fault" is a faster way of returning to real mode on many 80286 machines than the standard keyboard controller reset

-----H-09-----

INT 09 C - IRQ1 - KEYBOARD DATA READY

Desc: this interrupt is generated when data is received from the keyboard.

This is normally a scan code (from either a keypress *or* a key release), but may also be an ACK or NAK of a command on AT-class keyboards.

Notes: this IRQ may be masked by setting bit 1 on I/O port 21h
 if the BIOS supports an enhanced (101/102-key) keyboard, it calls
 INT 15/AH=4Fh after reading the scan code (see #00006) from the
 keyboard and before further processing; all further processing uses
 the scan code returned from INT 15/AH=4Fh
 the default interrupt handler is at F000h:E987h in 100%-compatible
 BIOSes
 the interrupt handler performs the following actions for certain
 special keystrokes:

- Ctrl-Break clear keyboard buffer, place word 0000h in buffer,
 invoke INT 1B, and set flag at 0040h:0071h
- SysReq invoke INT 15/AH=85h (SysReq is often labeled SysRq)
- Ctrl-Numlock place system in a tight wait loop until next INT 09
- Ctrl-Alt-Del jump to BIOS startup code (either F000h:FFF0h or the
 destination of the jump at that address)
- Shift-PrtSc invoke INT 05
- Ctrl-Alt-Plus (HP Vectra) enable keyclick
- Ctrl-Alt-Plus (many clones) set clock speed to high
- Ctrl-Alt-Minus (HP Vectra) reduce keyclick volume
- Ctrl-Alt-Minus (many clones) set clock speed to low
- Ctrl-Alt-SysReq (HP Vectra) generate hard reset
- Ctrl-Alt-S (many clones) run BIOS setup program
- Ctrl-Alt-Esc (many clones) run BIOS setup program
- Ctrl-Alt-Ins (many clones) run BIOS setup program
- Ctrl-Alt-LeftShift-GrayMinus (some clones) turn off system cache
- Ctrl-Alt-LeftShift-GrayPlus (some clones) turn on system cache

DR DOS hooks this interrupt to control the cursor shape (underscore/
 half block) for overwrite/insert mode

DR Multiuser DOS hooks this interrupt for cursor shape control and to
 control whether Ctrl-Alt-Del reboots the current session or the
 entire system

SeeAlso: INT 05"PRINT SCREEN",INT 0B"HP 95LX",INT 15/AH=4Fh,INT 15/AH=85h
 SeeAlso: INT 16/AH=00h,INT 16/AH=10h,INT 1B,INT 2F/AX=A901h,INT 4A/AH=00h"TI"
 SeeAlso: INT 51"DESQview",INT 59"DoubleDOS",INT 79"GO32"

(Table 00006)

Values for keyboard make/break (scan) code:

01h	Esc	31h	N		
02h	1 !	32h	M		
03h	2 @	33h	, <	63h	F16
04h	3 #	34h	. >	64h	F17


```

05h 4 $      35h / ?      65h F18
06h 5 %      36h Right Shift  66h F19
07h 6 ^      37h Grey*      67h F20
08h 7 &      38h Alt      68h F21 (Fn) [*]
09h 8 *      39h SpaceBar  69h F22
0Ah 9 (      3Ah CapsLock  6Ah F23
0Bh 0 )      3Bh F1      6Bh F24
0Ch - _      3Ch F2      6Ch --
0Dh = +      3Dh F3      6Dh EraseEOF
0Eh Backspace 3Eh F4
0Fh Tab      3Fh F5      6Fh Copy/Play
10h Q      40h F6
11h W      41h F7
12h E      42h F8      72h CrSel
13h R      43h F9      73h <delta> [*]
14h T      44h F10     74h ExSel
15h Y      45h NumLock  75h --
16h U      46h ScrollLock 76h Clear
17h I      47h Home      77h [Note2] Joyst But1
18h O      48h UpArrow  78h [Note2] Joyst But2
19h P      49h PgUp     79h [Note2] Joyst Right
1Ah [ {      4Ah Grey-    7Ah [Note2] Joyst Left
1Bh ] }      4Bh LeftArrow 7Bh [Note2] Joyst Up
1Ch Enter    4Ch Keypad 5 7Ch [Note2] Joyst Down
1Dh Ctrl     4Dh RightArrow 7Dh [Note2] right mouse
1Eh A      4Eh Grey+    7Eh [Note2] left mouse
1Fh S      4Fh End
20h D      50h DownArrow
21h F      51h PgDn
22h G      52h Ins
23h H      53h Del
24h J      54h SysReq   ---non-key codes---
25h K      55h [Note1] F11 00h kbd buffer full
26h L      56h left \ | (102-key)
27h ; :      57h F11     AAh self-test complete
28h ' "      58h F12     E0h prefix code
29h ` ~      59h [Note1] F15 E1h prefix code
2Ah Left Shift 5Ah PA1    EEh ECHO
2Bh \ |      5Bh F13 (LWin) F0h prefix code (key break)
2Ch Z      5Ch F14 (RWin) FAh ACK
2Dh X      5Dh F15 (Menu) FCh diag failure (MF-kbd)

```

```
2Eh C      FDh  diag failure (AT-kbd)
2Fh V      FEh  RESEND
30h B      FFh  kbd error/buffer full
```

Notes: scan codes 56h-E1h are only available on the extended (101/102-key)

keyboard and Host Connected (122-key) keyboard; scan codes 5Bh-5Dh are only available on the 122-key keyboard and the Microsoft Natural Keyboard; scan codes 5Eh-76h are only available on the 122-key keyboard

in the default configuration, break codes are the make scan codes with the high bit set; make codes 60h,61h,70h, etc. are not available because the corresponding break codes conflict with prefix codes (code 2Ah is available because the self-test result code AAh is only sent on keyboard initialization). An alternate keyboard configuration can be enabled on AT and later systems with enhanced keyboards, in which break codes are the same as make codes, but prefixed with an F0h scan code

prefix code E0h indicates that the following make/break code is for a "gray" duplicate to a key which existed on the original PC keyboard; prefix code E1h indicates that the following make code has no corresponding break code (currently only the Pause key generates no break code)

the Microsoft Natural Keyboard sends make codes 5Bh, 5Ch, and 5Dh (all with an E0h prefix) for the Left Windows, Right Windows, and Menu keys on the bottom row

the European "Cherry G81-3000 SAx/04" keyboard contains contacts for four additional keys, which can be made available by a user modification; the three new keys located directly below the cursor pad's Delete, End, and PgDn keys send make codes 66h-68h (F19-F21); the fourth new key, named <delta>, sends make code 73h

the SysReq key is often labeled SysRq

the "Accord" ergonomic keyboard with optional touchpad (no other identification visible on keyboard or in owner's booklet) has an additional key above the Grey- key marked with a left-pointing triangle and labeled "Fn" in the owner's booklet which returns scan codes E0h 68h on make and E0h E8h on break

the "Preh Commander AT" keyboard with additional F11-F22 keys treats F11-F20 as Shift-F1..Shift-F10 and F21/F22 as Ctrl-F1/Ctrl-F2; the Eagle PC-2 keyboard with F11-F24 keys treated those additional keys in the same way

[Note1] the "Cherry G80-0777" keyboard has additional F11-F15 keys which generate make codes 55h-59h; some other extended keyboards

generate codes 55h and 56h for F11 and F12, which cannot be managed by standard DOS keyboard drivers

[Note2] the Schneider/Amstrad PC1512 PC keyboards contain extra keys, a mouse, and a digital joystick, which are handled like extra keys. The joystick's motion scancodes are converted into standard arrow keys by the BIOS, and the joystick and mouse button scan codes are converted to FFFFh codes in the BIOS keyboard buffer (see CMOS 15h"AMSTRAD").

In addition to the keys listed in the table above, there are

Del-> (delete forward) 70h
Enter 74h

SeeAlso: #00602 at INT 16/AX=6F07h,#03214 at INT 4A/AH=05h

-----C-09-----

INT 09 C - CPU-generated (80286,80386) - PROCESSOR EXTENSION PROTECTION ERROR

Desc: called if the coprocessor attempts to access memory outside a segment boundary; it may occur at an arbitrary time after the coprocessor instruction was issued

Notes: until the condition is cleared or the coprocessor is reset, the only coprocessor instruction which may be used is FNINIT; WAIT or other coprocessor instructions will cause a deadlock because the coprocessor is still busy waiting for data

for the 486+, a coprocessor protection error generates an INT 0D

SeeAlso: INT 07"CPU"

-----*-09-----

INT 09 P - internal hardware - RESERVED BY Intel (80486)

Note: this exception has been moved to INT 0D

SeeAlso: INT 09"CPU",INT 0D

-----H-0A-----

INT 0A C - IRQ2 - LPT2 (PC), VERTICAL RETRACE INTERRUPT (EGA,VGA)

Notes: the TOPS and PCnet adapters use this interrupt request line by default DOS 3.2 revector IRQ2 to a stack-switching routine; DOS 3.3+ does so unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS 3.3+ use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ on ATs and above, the physical data line for IRQ2 is labeled IRQ9 and connects to the slave 8259. The BIOS redirects the interrupt for IRQ9 back here.

under DESQview, only the INT 15h vector and BASIC segment address (the word at 0000h:0510h) may be assumed to be valid for the handler's process

many VGA boards do not implement the vertical retrace interrupt, including the IBM VGA Adapter where the traces are either cut or

removed

SeeAlso: INT 52"DESQview",INT 5A"DoubleDOS",INT 71,INT 7A"GO32"

-----H-0A-----

INT 0A C - IRQ2 - Tandy 1000-series HARD DISK

Notes: this interrupt may be masked by setting bit 2 on I/O port 21h

the Tandy 1000, 1000A, and 1000HD use IRQ2 for the hard disk; the 1000EX, HX, RLX, RLX-HD, RLX-B, RLX-HD-B use IRQ5 instead; the 1000RL, RL-HD, SL, SL/2, TL, TL/2, and TL/3 are jumper-selectable for either IRQ2 or IRQ5 (default IRQ5); the 1000SX and TX are DIP-switch selectable for IRQ2 or IRQ5 (default IRQ2); the RSX and RSX-HD use IRQ14. Tandy systems which use IRQ2 for the hard disk interrupt use IRQ5 for vertical retrace.

SeeAlso: INT 52"DESQview",INT 5A"DoubleDOS",INT 71

-----H-0A-----

INT 0A - IRQ2 - ROLAND MPU MIDI INTERFACE

Note: newer Roland cards and MIDI interfaces by other manufacturers use a jumper-selectable IRQ, but software and hardware generally defaults to IRQ2

SeeAlso: INT 52"DESQview",INT 5A"DoubleDOS",INT 71,INT 7A"GO32"

-----C-0A-----

INT 0A CP - CPU-generated (80286+) - INVALID TASK STATE SEGMENT

Desc: automatically called during a task switch if the new TSS specified by

the task gate is invalid for any of the following reasons:

TSS limit is less than 43 (80286) or 103 (80386/80486)

LDT selector invalid or segment not present

null SS selector, or SS selector outside LDT/GDT limit

stack segment is read-only

stack segment DPL differs from new CPL, or RPL <> CPL

CS selector is outside LDT/GDT limit or not code

non-conforming code segment's DPL differs from CPL

conforming code segment's DPL > CPL

DS/ES selectors outside LDT/GDT limit or not readable segments

Note: the handler must use a task gate in order to have a valid TSS under which to execute; it must also reset the busy bit in the new TSS

SeeAlso: INT 0B"CPU"

-----H-0B-----

INT 0B C - IRQ3 - SERIAL COMMUNICATIONS (COM2)

Desc: automatically asserted by the UART when COM2 needs attention, if the UART has been programmed to generate interrupts

Notes: the TOPS and PCnet adapters use this interrupt request line as an alternate

on PS/2s, COM2 through COM8 share this interrupt; on many PC's, COM4 shares this interrupt
may be masked by setting bit 3 on I/O port 21h
DOS 3.3+ revector IRQ3 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ
SeeAlso: INT 0C"COM1",INT 53"DESQview",INT 5B"DoubleDOS",INT 7B"GO32"

-----C-0B-----

INT 0B CP - CPU-generated (80286+) - SEGMENT NOT PRESENT

Desc: generated when loading a segment register if the segment descriptor indicates that the segment is not currently in memory, unless the segment is an LDT (see INT 0A"CPU") or stack segment (see INT 0C"CPU") needed by a task switch

Note: may be used to implement virtual memory by loading in segments as they are accessed, clearing the "not present" bit after loading

SeeAlso: INT 0A"CPU",INT 0E"CPU"

-----h-0B-----

INT 0B - HP 95LX - LOW-LEVEL KEYBOARD HANDLER

Desc: this is the lowest-level keyboard handler on an HP 95LX palmtop

Note: debounces key, places the keycode in I/O register 60h, and calls INT 09

SeeAlso: INT 09,INT 0D"HP 95LX",INT 15/AX=4DD4h

-----H-0C-----

INT 0C C - IRQ4 - SERIAL COMMUNICATIONS (COM1)

Desc: automatically asserted by the UART when COM1 needs attention, if the UART has been programmed to generate interrupts

BUG: this vector is modified but not restored by Direct Access v4.0, and may be left dangling by other programs written with the same version of compiled BASIC

Notes: on many PC's, COM3 shares this interrupt
may be masked by setting bit 4 on I/O port 21h

DOS 3.3+ revector IRQ4 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ

SeeAlso: INT 0B"COM2",INT 54"DESQview",INT 5C"DoubleDOS",INT 7C"GO32"

-----C-0C-----

INT 0C C - CPU-generated (80286+) - STACK FAULT

Desc: this interrupt is generated in protected mode on a stack overflow or underflow, or if an inter-level transition or task switch references a stack segment marked "not present"; it is generated in real mode on accessing a word operand at SS:FFFFh

Note: the 80286 will shut down in real mode if SP=1 before a push. On the

PC AT and compatibles, external circuitry generates a reset on shutdown.

SeeAlso: INT 0B"CPU",INT 0D"CPU"

-----I-0C-----

INT 0C - IBM SYSTEM 36/38 WORKSTATION EMULATION - API POINTER

Desc: the IBM System 36/38 emulator may be invoked through a private API, whose entry point address (see #00007) is offset 100h in the segment pointed at by this vector

(Table 00007)

Call the System 36/38 emulator API entry point with:

AH = function

03h update screen

05h select next session

AL = session number (00h-03h)

Return: AL = session type code

00h not active

01h display session

02h printer session

FEh invalid session number

DS = requested session's data segment (0 if not active)

(see #00008)

Return: ???

Format of System 36/38 emulator's data area:

Offset Size Description (Table 00008)

13Eh BYTE bit flags for status line indicators turned on since this byte last zerod

13Fh BYTE bit flags for status line indicators turned off since this byte last set to FFh

140h WORD offset of EBCDIC to ASCII translation

146h WORD offset of EBCDIC screen buffer

148h WORD offset of EC (engineering change) level signature

150h BYTE "KEYI"

151h BYTE 5250 key scan code to be sent to remote

15Bh BYTE "SYSAB"

15Dh BYTE 5250 cursor column

15Eh BYTE 5250 cursor row

167h BYTE "DVCTAD"

178h BYTE "FLAGS"

184h BYTE "SESSNOAD"

193h BYTE "STNAD"

198h BYTE "NSDS"

Note: offsets are from the interrupt handler's segment

-----H-0D-----

INT 0D C - IRQ5 - FIXED DISK (PC,XT), LPT2 (AT), reserved (PS/2)

Notes: under DESQview, only the INT 15h vector and BASIC segment address (the word at 0000h:0510h) may be assumed to be valid for the handler's process

DOS 3.3+ revector IRQ5 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ

the Tandy 1000, 1000A, and 1000HD use IRQ2 for the hard disk; the 1000EX, HX, RLX, RLX-HD, RLX-B, RLX-HD-B use IRQ5 instead; the 1000RL, RL-HD, SL, SL/2, TL, TL/2, and TL/3 are jumper-selectable for either IRQ2 or IRQ5 (default IRQ5); the 1000SX and TX are DIP-switch selectable for IRQ2 or IRQ5 (default IRQ2); the RSX and RSX-HD use IRQ14. Tandy systems which use IRQ2 for the hard disk interrupt use IRQ5 for vertical retrace.

may be masked by setting bit 5 on I/O port 21h

SeeAlso: INT 0E"IRQ6",INT 0F"IRQ7",INT 55"DESQview",INT 5D"DoubleDOS"

SeeAlso: INT 7D"GO32"

-----H-0D-----

INT 0D C - IRQ5 - Tandy 1000 60 Hz RAM REFRESH

Desc: used to ensure that the dynamic RAM retains its contents in earlier 1000-series models; later models use normal DMA for RAM refresh (some early models have no DMA unless it is added via an expansion card)

SeeAlso: INT 55

-----h-0D-----

INT 0D - HP 95LX - INFRARED INTERRUPT

Desc: this interrupt is used to perform communications over the infrared data link

SeeAlso: INT 0B"HP 95LX",INT 0E"HP 95LX",INT 15/AX=4DD4h

-----C-0D-----

INT 0D C - CPU-generated (80286+) - GENERAL PROTECTION VIOLATION

Desc: the CPU generates this interrupt when it detects a protection violation which does not fit under any other category having a separate interrupt

Notes: called in real mode when

an instruction accesses a memory operand extending beyond offset FFFFh (i.e. WORD at FFFFh or DWORD at FFFDh or higher) in segment

CS, DS, ES, FG, or GS
 a PUSH MEM or POP MEM instruction contains an invalid bit encoding
 in the second byte
 an instruction exceeds the maximum length allowed (10 bytes for
 80286, 15 bytes for 80386/80486)
 an instruction wraps from offset FFFFh to offset 0000h
 called in protected mode on protection violations not covered by INT 06
 through INT 0C, including
 segment limit violations
 write to read-only segments
 accesses using null DS or ES selectors
 accesses to segments with privilege greater than CPL
 wrong descriptor type
 called on 80486 protected-mode floating-point protection fault
 SeeAlso: INT 09"80486",INT 0C"STACK"

-----H-0E-----

INT 0E C - IRQ6 - DISKETTE CONTROLLER

Desc: this interrupt is generated by the floppy disk controller on
 completion of an operation

Notes: default handler is at F000h:EF57h in IBM PC and 100%-compatible BIOSes
 may be masked by setting bit 6 on I/O port 21h

DOS 3.3+ revector IRQ6 to a stack-switching routine unless STACKS=0
 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing
 Protocol (see #02568) when hooking this IRQ

SeeAlso: INT 0D"IRQ5",INT 56"DESQview",INT 5E"DoubleDOS",INT 7E"GO32"

-----C-0E-----

INT 0E C - CPU-generated (80386+ native mode) - PAGE FAULT

Desc: this interrupt is generated in protected and virtual-86 modes on
 attempting to access a 4K memory page whose page table entry has
 the "present" bit cleared

Notes: used to implement virtual memory--when the page fault occurs, the
 operating system can load the appropriate page from disk

a protected-mode program written in the Flat memory model will often
 generate this interrupt (typically reported as an Exception #14 or
 Exception 0E with a register dump) when dereferencing an
 uninitialized or corrupted pointer

SeeAlso: INT 0B"CPU"

-----h-0E-----

INT 0E C - HP 95LX - EXTERNAL CARD INTERRUPT

SeeAlso: INT 0D"HP 95LX",INT 0F"HP 95LX",INT 15/AX=4DD4h

-----H-0F-----

INT 0F C - IRQ7 - PARALLEL PRINTER

Desc: this interrupt is generated by the LPT1 printer adapter when the printer becomes ready

Notes: most printer adapters do not reliably generate this interrupt the 8259 interrupt controller generates an interrupt corresponding to IRQ7 when an error condition occurs

SeeAlso: INT 0D"IRQ5",INT 57"DESQview",INT 5F"DoubleDOS",INT 7F"GO32"

SeeAlso: MEM 0040h:006Bh

-----h-0F-----

INT 0F C - HP 95LX - REAL-TIME CLOCK

SeeAlso: INT 0E"HP 95LX",INT 15/AX=4DD4h,INT 70"IRQ8"

-----C-10-----

INT 10 C - CPU-generated (80286+) - COPROCESSOR ERROR

Desc: this interrupt is generated by the CPU when the -ERROR pin is asserted by the coprocessor

Note: AT's and clones usually wire the coprocessor to use IRQ13, but not all get it right

SeeAlso: INT 09"hardware",INT 75

-----V-10-----

INT 10h - LIRVGA19 - CHAR HEIGHT HOOK

Desc: LIRVGA19.COM fixes display problems on some notebooks by hooking INT 10h and resetting the BIOS character height (MEM 0040h:0085h) to 10h on return, if it happens to be greater than 10h.

Program: LIRVGA19.COM is part of the Lithuanian LIR codepage support package developed by LABAS u.a.b.

SeeAlso: INT 10/AX=5555h/BX=BBBBh,MEM 0040h:0085h

-----V-1000-----

INT 10 - VIDEO - SET VIDEO MODE

AH = 00h

AL = desired video mode (see #00010)

Return: AL = video mode flag (Phoenix, AMI BIOS)

20h mode > 7

30h modes 0-5 and 7

3Fh mode 6

AL = CRT controller mode byte (Phoenix 386 BIOS v1.10)

Desc: specify the display mode for the currently active display adapter

InstallCheck: for Ahead adapters, the signature "AHEAD" at C000h:0025h

for Paradise adapters, the signature "VGA=" at C000h:007Dh

for Oak Tech OTI-037/057/067/077 chipsets, the signature "OAK VGA" at C000h:0008h

for ATI adapters, the signature "761295520" at C000h:0031h; the byte

at C000h:0043h indicates the chipset revision:
31h for 18800
32h for 18800-1
33h for 18800-2
34h for 18800-4
35h for 18800-5
62h for 68800AX (Mach32) (see also #00732)
the two bytes at C000h:0040h indicate the adapter type

"22" EGA Wonder

"31" VGA Wonder

"32" EGA Wonder800+

the byte at C000h:0042h contains feature flags

bit 1: mouse port present

bit 4: programmable video clock

the byte at C000h:0044h contains additional feature flags if chipset
byte > 30h (see #00009)

for Genoa video adapters, the signature 77h XXh 99h 66h at C000h:NNNNh,
where NNNNh is stored at C000h:0037h and XXh is

00h for Genoa 6200/6300

11h for Genoa 6400/6600

22h for Genoa 6100

33h for Genoa 5100/5200

55h for Genoa 5300/5400

for SuperEGA BIOS v2.41+, C000h:0057h contains the product level

for Genoa SuperEGA BIOS v3.0+, C000h:0070h contains the signature

"EXTMODE", indicating support for extended modes

Notes: IBM standard modes do not clear the screen if the high bit of AL is set
(EGA or higher only)

the Tseng ET4000 chipset is used by the Orchid Prodesigner II, Diamond
SpeedSTAR VGA, Groundhog Graphics Shadow VGA, Boca Super X VGA,
Everex EV-673, etc.

intercepted by GRAFTABL from Novell DOS 7 and Caldera OpenDOS 7.01.

SeeAlso: AX=0070h,AX=007Eh,AX=10E0h,AX=10F0h,AH=40h,AX=6F05h,AH=FFh"GO32"

SeeAlso: INT 33/AX=0028h,INT 5F/AH=00h,INT 62/AX=0001h,MEM 0040h:0049h

Index: installation check;Tseng ET4000|installation check;Ahead video cards

Index: installation check;Oak Technologies|installation check;ATI video cards

Index: installation check;Paradise video|installation check;Genoa video cards

Bitfields for ATI additional feature flags:

Bit(s) Description (Table 00009)

0 70 Hz non-interlaced display

1 Korean (double-byte) characters
 2 45 MHz memory clock rather than 40 MHz
 3 zero wait states
 4 paged ROMs
 6 no 8514/A monitor support
 7 HiColor DAC

(Table 00010)

Values for video mode:

	text/ grph	text resol	pixel box	pixel resolution	colors	disply pages	scrn addr	system
00h	= T	40x25	8x8	320x200	16gray	8	B800	CGA, PCjr, Tandy
	= T	40x25	8x14	320x350	16gray	8	B800	EGA
	= T	40x25	8x16	320x400	16	8	B800	MCGA
	= T	40x25	9x16	360x400	16	8	B800	VGA
01h	= T	40x25	8x8	320x200	16	8	B800	CGA, PCjr, Tandy
	= T	40x25	8x14	320x350	16	8	B800	EGA
	= T	40x25	8x16	320x400	16	8	B800	MCGA
	= T	40x25	9x16	360x400	16	8	B800	VGA
02h	= T	80x25	8x8	640x200	16gray	4	B800	CGA, PCjr, Tandy
	= T	80x25	8x14	640x350	16gray	8	B800	EGA
	= T	80x25	8x16	640x400	16	8	B800	MCGA
	= T	80x25	9x16	720x400	16	8	B800	VGA
03h	= T	80x25	8x8	640x200	16	4	B800	CGA, PCjr, Tandy
	= T	80x25	8x14	640x350	16/64	8	B800	EGA
	= T	80x25	8x16	640x400	16	8	B800	MCGA
	= T	80x25	9x16	720x400	16	8	B800	VGA
	= T	80x43	8x8	640x350	16	4	B800	EGA, VGA [17]
	= T	80x50	8x8	640x400	16	4	B800	VGA [17]
04h	= G	40x25	8x8	320x200	4	.	B800	CGA, PCjr, EGA, MCGA, VGA
05h	= G	40x25	8x8	320x200	4gray	.	B800	CGA, PCjr, EGA
	= G	40x25	8x8	320x200	4	.	B800	MCGA, VGA
06h	= G	80x25	8x8	640x200	2	.	B800	CGA, PCjr, EGA, MCGA, VGA
	= G	80x25	.	. mono	.	.	B000	HERCULES.COM on HGC [14]
07h	= T	80x25	9x14	720x350	mono	var	B000	MDA, Hercules, EGA
	= T	80x25	9x16	720x400	mono	.	B000	VGA
08h	= T	132x25	8x8	1056x200	16	.	B800	ATI EGA/VGA Wonder [2]
	= T	132x25	8x8	1056x200	mono	.	B000	ATI EGA/VGA Wonder [2]
	= G	20x25	8x8	160x200	16	.	.	PCjr, Tandy 1000
	= G	80x25	8x16	640x400	color	.	.	Tandy 2000
	= G	90x43	8x8	720x348	mono	.	B000	Hercules + MSHERC.COM

```

= G 90x45 8x8 720x360 mono . B000 Hercules + HERKULES [11]
= G 90x29 8x12 720x348 mono . . Hercules + HERCBIOS [15]
09h = G 40x25 8x8 320x200 16 . . PCjr, Tandy 1000
= G 80x25 8x16 640x400 mono . . Tandy 2000
= G 90x43 8x8 720x348 mono . . Hercules + HERCBIOS [15]
0Ah = G 80x25 8x8 640x200 4 . . PCjr, Tandy 1000
0Bh = reserved (EGA BIOS internal use)
= G 80x25 8x8 640x200 16 . . Tandy 1000 SL/TL [13]
0Ch = reserved (EGA BIOS internal use)
0Dh = G 40x25 8x8 320x200 16 8 A000 EGA,VGA
0Eh = G 80x25 8x8 640x200 16 4 A000 EGA,VGA
0Fh = G 80x25 8x14 640x350 mono 2 A000 EGA,VGA
10h = G 80x25 8x14 640x350 4 2 A000 64k EGA
= G . . 640x350 16 . A000 256k EGA,VGA
11h = G 80x30 8x16 640x480 mono . A000 VGA,MCGA,ATI EGA,ATI VIP
12h = G 80x30 8x16 640x480 16/256K . A000 VGA,ATI VIP
= G 80x30 8x16 640x480 16/64 . A000 ATI EGA Wonder
= G . . 640x480 16 . . UltraVision+256K EGA
13h = G 40x25 8x8 320x200 256/256K . A000 VGA,MCGA,ATI VIP
14h = T 132x25 Nx16 . 16 . B800 XGA, IBM Enhanced VGA [3]
= T 132x25 8x16 1056x400 16/256K . . Cirrus CL-GD5420/5422/5426
= G 80x25 8x8 640x200 . . . Lava Chrome II EGA
= G . . 640x400 16 . . Tecmar VGA/AD
15h = G 80x25 8x14 640x350 . . . Lava Chrome II EGA
16h = G 80x25 8x14 640x350 . . . Lava Chrome II EGA
= G . . 800x600 16 . . Tecmar VGA/AD
17h = T 132x25 . . . . . Tecmar VGA/AD
= T 80x43 8x8 640x348 16 4 B800 Tseng ET4000 BIOS [10]
= G 80x34 8x14 640x480 . . . Lava Chrome II EGA
18h = T 80x30 9x16 720x480 16 1 A000 Realtek RTVGA [12]
= T 132x25 . . mono . B000 Cirrus 5320 chipset
= T 132x44 8x8 1056x352 mono . B000 Tseng Labs EVA
= T 132x44 9x8 1188x352 4gray 2 B000 Tseng ET3000 chipset
= T 132x44 8x8 1056x352 16/256 2 B000 Tseng ET4000 chipset
= G 80x34 8x14 640x480 . . . Lava Chrome II EGA
= G 1024x768 16 . . Tecmar VGA/AD
19h = T 80x43 9x11 720x473 16 1 A000 Realtek RTVGA [12]
= T 132x25 8x14 1056x350 mono . B000 Tseng Labs EVA
= T 132x25 9x14 1188x350 4gray 4 B000 Tseng ET3000 chipset
= T 132x25 8x14 1056x350 16/256 4 B000 Tseng ET4000 chipset
= T 132x34 . . mono . B000 Cirrus 5320 chipset

```

```

1Ah = T 80x60 9x8 720x480 16 1 A000 Realtek RTVGA [12]
    = T 132x28 8x13 1056x364 mono . B000 Tseng Labs EVA
    = T 132x28 9x13 1188x364 4gray 4 B000 Tseng ET3000 chipset
    = T 132x28 8x13 1056x364 16/256 4 B000 Tseng ET4000 chipset
    = T 132x44 . . mono . B000 Cirrus 5320 chipset
    = G . . 640x350 256 . . Tecmar VGA/AD
1Bh = T 132x25 9x14 1188x350 16 1 A000 Realtek RTVGA [12]
    = G . . 640x400 256 . . Tecmar VGA/AD
1Ch = T 132x25 . . . . Cirrus 5320 chipset
    = T 132x30 9x16 1188x480 16 1 A000 Realtek RTVGA [12]
    = G . . 640x480 256 . . Tecmar VGA/AD
1Dh = T 132x43 . . . . Cirrus 5320 chipset
    = T 132x43 9x11 1188x473 16 1 A000 Realtek RTVGA [12]
    = G . . 800x600 256 . . Tecmar VGA/AD
1Eh = T 132x44 . . . . Cirrus 5320 chipset
    = T 132x60 9x8 1188x480 16 1 A000 Realtek RTVGA [12]
1Fh = G 100x75 8x8 800x600 16 1 A000 Realtek RTVGA
20h = T 132x25 . . 16 . . Avance Logic AL2101
    = G 40x16 . 240x128 mono . B000 HP 95LX/100LX/200LX
    = G 80x30 8x16 640x480 16 . . C&T 64310/65530 BIOS
    = G 120x45 8x16 960x720 16 1 A000 Realtek RTVGA
21h = T 80x25 . . mono . B000 HP 200LX
    = T 132x30 . . 16 . . Avance Logic AL2101
    = T 132x44 9x9 1188x396 16/256K . B800 WD90C
    = T 132x44 9x9 1188x396 16 . B800 Diamond Speedstar 24X
    = T 132x60 . . 16 2 B800 Tseng ET4000 chipset [10]
    = G 80x43 8x8 720x348 mono . B000 DESQview 2.x+Hercules [4]
    = G 128x48 8x16 1024x768 16 1 A000 Realtek RTVGA [12]
22h = T 132x43 . . . . Allstar Peacock (VGA)
    = T 132x43 . . 16 . . Avance Logic AL2101
    = T 132x44 8x8 1056x352 . . B800 Tseng Labs EVA
    = T 132x44 9x8 1188x352 16/256K 2 B800 Tseng ET3000 chipset
    = T 132x44 8x8 1056x352 16/256K 2 B800 Tseng ET4000 chipset
    = T 132x44 8x8 1056x352 . . Ahead Systems EGA2001
    = T 132x44 8x8 1056x352 16 2 B800 Ahead B
    = T 132x44 8x9 1056x398 16 . . STB Lightspeed ET4000/W32P
    = T 132x44 . . 16 . . Orchid Prodesigner VGA
    = G 80x43 8x8 720x348 mono . B800 DESQview 2.x+Hercules [4]
    = G 96x64 8x16 768x1024 16 1 A000 Realtek RTVGA
    = G 100x37 8x16 800x600 16 . . C&T 64310/65530 BIOS
23h = T 132x25 6x14 792x350 . . B800 Tseng Labs EVA

```

```

= T 132x25 9x14 1188x350 16/256K 4 B800 Tseng ET3000 chipset
= T 132x25 8x14 1056x350 16/256 4 B800 Tseng ET4000 chipset
= T 132x25 8x14 1056x350 . . . Ahead Systems EGA2001
= T 132x25 8x14 1056x350 16 4 B800 Ahead B
= T 132x25 8x8 1056x200 16 . B800 ATI EGA Wonder,ATI VIP
= T 132x25 . . . . . Cirrus 5320 chipset
= T 132x28 . . . . . Allstar Peacock (VGA)
= T 132x28 . . 16 . . Orchid Prodesigner VGA
= T 132x60 . . 16 . . Avance Logic AL2101
= G 128x48 8x16 1024x768 4 1 A000 Realtek RTVGA
24h = T 80x30 . . 16 . . Avance Logic AL2101
= T 132x25 . . . . . Allstar Peacock (VGA)
= T 132x25 . . 16 . . Orchid Prodesigner VGA
= T 132x28 6x13 792x364 . . B800 Tseng Labs EVA
= T 132x28 9x13 1188x364 16/256K 4 B800 Tseng ET3000 chipset
= T 132x28 8x12 1056x336 16 1 B800 Ahead B
= T 132x28 8x13 1056x364 16/256K 4 B800 Tseng ET4000 chipset
= T 132x28 8x14 1056x392 16 . . STB Lightspeed ET4000/W32P
= T 132x28 . . . . . Cirrus 5320 chipset
= G 64x32 8x16 512x512 256 1 A000 Realtek RTVGA
= G 128x48 8x16 1024x768 16 . . C&T 64310/65530 BIOS
25h = T 80x43 . . 16 . . Avance Logic AL2101
= G 80x60 8x8 640x480 . . A000 Tseng Labs EVA
= G 80x60 8x8 640x480 16/256K 1 A000 Tseng ET3000/4000 chipset
= G . . 640x480 16 . . VEGA VGA
= G 80x60 8x8 640x480 16 . A000 Orchid Prodesigner VGA
= G 80x60 8x8 640x480 16 1 A000 Ahead B (same as 26h)
= G . . 640x480 16 . . NEC GB-1
= G . . 640x480 16 . . Cirrus 5320 chipset
= G . . 640x400 256 . . Realtek RTVGA
26h = T 80x60 8x8 640x480 . . . Tseng Labs EVA
= T 80x60 8x8 640x480 16/256K 3 B800 Tseng ET3000/4000 chipset
= T 80x60 . . . . . Allstar Peacock (VGA)
= T 80x60 . . 16 . . Orchid ProDesigner VGA
= T 80x60 . . 16 . . Avance Logic AL2101
= G 80x60 8x8 640x480 . . . Ahead Systems EGA2001
= G 80x60 8x8 640x480 16 1 A000 Ahead B (same as 25h)
= G . . 640x480 256 . . Realtek RTVGA
27h = T 132x25 8x8 1056x200 mono . B000 ATI EGA Wonder,ATI VIP
= G . . 720x512 16 . . VEGA VGA
= G . . 720x512 16 . . Genoa

```

```

= G 100x75 8x8 800x600 256 1 A000 Realtek RTVGA [12]
= G . . 960x720 16 . . Avance Logic AL2101
28h = T ???x??? . . . . VEGA VGA
= G . . 512x512 256 . . Avance Logic AL2101
= G . . 1024x768 256 . . Realtek RTVGA (1meg)
= G 160x64 8x16 1280x1024 16 . . Chips&Technologies 64310 [1]
29h = G . . 640x400 256 . . Avance Logic AL2101
= G . . 800x600 16 . . VEGA VGA
= G 100x37 8x16 800x600 16 . A000 Orchid
= G . . 800x600 16 . A000 STB,Genoa,Sigma
= G . . 800x600 16 . . Allstar Peacock (VGA)
= G 100x37 8x16 800x600 16/256K 1 A000 Tseng ET3000/4000 chipset
= G . . 800x600 ??? . . EIZO MDB10
= G . . 800x600 16 . . Cirrus 5320 chipset
= G NA . 800x600 16 . . Compaq QVision 1024/1280
= G . . 1024x1024 256 . . Realtek RTVGA BIOS v3.C10
2Ah = T 100x40 . . . . Allstar Peacock (VGA)
= T 100x40 8x16 800x640 16 . . Orchid Prodesigner VGA
= T 100x40 8x15 800x600 16/256K 4 B800 Tseng ET3000/4000 chipset
= T 100x40 8x15 800x600 16 . . STB Lightspeed ET4000/W32P
= G . . 640x480 256 . . Avance Logic AL2101
= G . . 1280x1024 16 . . Realtek RTVGA
2Bh = G . . 800x600 16 . . Avance Logic AL2101
2Ch = G . . 800x600 256 . . Avance Logic AL2101
2Dh = G . . 640x350 256 . . VEGA VGA
= G . . 640x350 256/256K . A000 Orchid, Genoa, STB
= G 80x25 8x14 640x350 256/256K 1 A000 Tseng ET3000/4000 chipset
= G . . 640x350 256 . . Cirrus 5320 chipset
= G 80x25 8x14 640x350 256 . . STB Lightspeed ET4000/W32P
= G . . 768x1024 16 . . Avance Logic AL2101
2Eh = G . . 640x480 256 . . VEGA VGA
= G 80x30 8x16 640x480 256/256K . A000 Orchid
= G . . 640x480 256/256K . A000 STB,Genoa,Sigma
= G 80x30 8x16 640x480 256/256K 1 A000 Tseng ET3000/4000 chipset
= G . . 640x480 256/256K . . Compaq QVision 1024/1280
= G . . 768x1024 256 . . Avance Logic AL2101
2Fh = T 160x50 8x8 1280x400 16 4 B800 Ahead B (Wizard/3270)
= G . . 720x512 256 . . VEGA VGA
= G . . 720x512 256 . . Genoa
= G 80x25 8x16 640x400 256/256K 1 A000 Tseng ET4000 chipset
= G . . 1024x768 4 . . Avance Logic AL2101

```

```

30h = G 80x30 8x16 640x480 256 . . C&T 64310/65530 BIOS
      = G . . . . B800 AT&T 6300
      = G . . 720x350 2 . . 3270 PC
      = G . . 800x600 256 . . VEGA VGA
      = G 100x37 8x16 800x600 256/256K . A000 Orchid
      = G . . 800x600 256/256K . A000 STB,Genoa,Sigma
      = G . . 800x600 256 . . Cardinal
      = G 100x37 8x16 800x600 256/256K 1 A000 Tseng ET3000/4000 chipset
      = G . . 1024x768 16 . . Avance Logic AL2101
31h = G . . 1024x768 256 . . Avance Logic AL2101
32h = T 80x34 8x10 . 16 4 B800 Ahead B (Wizard/3270)
      = G . . 640x480 256 . . Compaq QVision 1024/1280
      = G 100x37 8x16 800x600 256 . . C&T 64310/65530 BIOS
33h = T 132x44 8x8 . 16 . B800 ATI EGA Wonder,ATI VIP
      = T 80x34 8x8 . 16 4 B800 Ahead B (Wizard/3270)
34h = T 80x66 8x8 . 16 4 B800 Ahead B (Wizard/3270)
      = G . . 800x600 256 . . Compaq QVision 1024/1280
      = G 128x48 8x16 1024x768 256 . . Chips&Technologies 64310
36h = G . . 960x720 16 . . VEGA VGA, STB
      = G . . 960x720 16 . A000 Tseng ET3000 only
      = G . . 1280x1024 16 . . Avance Logic AL2101
37h = T 132x44 8x8 . mono . B800 ATI EGA Wonder,ATI VIP
      = G . . 1024x768 16 . . VEGA VGA
      = G 128x48 8x16 1024x768 16 . A000 Orchid
      = G . . 1024x768 16 . A000 STB,Genoa,Sigma
      = G . . 1024x768 16 . . Definicon
      = G 128x48 8x16 1024x768 16 1 A000 Tseng ET3000/4000 chipset
      = G . . 1024x768 16 . . Compaq QVision 1024/1280
      = G . . 1280x1024 256 . . Avance Logic AL2101
38h = G . . 1024x768 256 . . STB VGA/EM-16 Plus (1MB)
      = G 128x48 8x16 1024x768 256/256K 1 A000 Tseng ET4000 chipset
      = G . . 1024x768 256 . . Orchid ProDesigner II
      = G . . 1024x768 256 . . Compaq QVision 1024/1280
      = G 160x64 8x16 1280x1024 256 . . Chips&Technologies 64310 [1]
39h = G . . 1280x1024 16 . . Compaq QVision 1280
3Ah = G . . 1280x1024 256 . . Compaq QVision 1280
3Bh = G . . 512x480 256 . . Compaq QVision 1024/1280
3Ch = G . . 640x400 64K . . Compaq QVision 1024/1280
3Dh = G . . 1280x1024 16 . . Definicon
      = G 128x64 8x16 1280x1024 16 1 A000 Tseng ET4000 v3.00 [1,7]
3Eh = G . . 1280x961 16 . . Definicon

```



```

= G      .      .      640x480  64K      . .  Compaq QVision 1024/1280
3Fh = G      .      .      1280x1024 256      . .  Hercules ??? (ET4000W32)
= G      .      .      800x600   64K      . .  Compaq QVision 1024/1280
40h = T 80x43   .      .      .      .      . .  VEGA VGA, Tecmar VGA/AD
= T 80x43   .      .      .      .      . .  Video7 V-RAM VGA
= T 80x43   .      .      .      .      . .  Tatung VGA
= T 100x30  .      .      16      .      . .  MORSE VGA
= T 100x30  .      .      .      .      . .  Cirrus 510/520 chipset
= T 80x25   .      720x350 mono      . .  Genoa SuperEGA BIOS 3.0+
= G      .      .      320x200  64K      . .  Avance Logic AL2101
= G 80x25   8x16   640x400   2   1   B800 AT&T 6300, AT&T VDC600
= G 80x25   8x16   640x400   2   1   B800 Olivetti Quaderno
= G 80x25   8x16   640x400   2   1   B800 Compaq Portable
= G 80x30   8x16   640x480  32K   . .  Chips&Technologies 64310
= G      .      .      1024x768  64K      . .  Compaq QVision 1280
41h = T 132x25 .      .      .      .      . .  VEGA VGA
= T 132x25 .      .      .      .      . .  Tatung VGA
= T 132x25 .      .      .      .      . .  Video7 V-RAM VGA
= T 100x50 .      .      16      .      . .  MORSE VGA
= T 100x50 .      .      .      .      . .  Cirrus 510/520 chipset
= T 80x34   9x14   720x476  16/256K .  B800 WD90C
= T 80x34   9x14   .      16      .      B800 Diamond Speedstar 24X
= G      .      .      512x512  64K      . .  Avance Logic AL2101
= G      .      .      640x200  16   1 .  AT&T 6300
= G 80x30   8x16   640x480   64K   . .  Chips&Technologies 64310
= G 80x25   .      720x348 mono      .  B000 Genoa SuperEGA BIOS 3.0+
42h = T 132x43 .      .      .      .      . .  VEGA VGA
= T 132x43 .      .      .      .      . .  Tatung VGA
= T 132x43 .      .      .      .      . .  Video7 V-RAM VGA
= T 80x34   9x10   .      4   4   B800 Ahead B (Wizard/3270)
= T 100x60 .      .      16      .      . .  MORSE VGA
= T 100x60 .      .      .      .      . .  Cirrus 510/520 chipset
= G 80x25   8x16   640x400  16   . .  AT&T 6300, AT&T VDC600
= G      .      .      640x400  64K      . .  Avance Logic AL2101
= G 80x25   .      720x348 mono      .  B800 Genoa SuperEGA BIOS 3.0+
= G 100x37  8x16   800x600  32K   . .  Chips&Technologies 64310
43h = T 80x60 .      .      .      .      . .  VEGA VGA
= T 80x60 .      .      .      .      . .  Tatung VGA
= T 80x60 .      .      .      .      . .  Video7 V-RAM VGA
= T 80x45   9x8    .      4   4   B800 Ahead B (Wizard/3270)
= T 100x75 .      .      16      .      . .  MORSE VGA

```

```

= T 80x29 . 720x348 mono . . Genoa SuperEGA BIOS 3.0+
= G . . 640x200 of 640x400 viewport AT&T 6300 (unsupported)
= G . . 640x480 64K . . Avance Logic AL2101
= G 100x37 8x16 800x600 64K . . Chips&Technologies 64310
44h = disable VDC and DEB output . AT&T 6300
= T 100x60 . . . . . VEGA VGA
= T 100x60 . . . . . Tatung VGA
= T 100x60 . . . . . Video7 V-RAM VGA
= T 80x32 . 720x352 mono . . Genoa SuperEGA BIOS 3.0+
= G . . 800x600 64K . . Avance Logic AL2101
45h = T 132x28 . . . . . Tatung VGA
= T 132x28 . . . . . Video7 V-RAM VGA
= T 80x44 . 720x352 mono . . Genoa SuperEGA BIOS 3.0+
46h = T 132x25 8x14 . mono . . Genoa 6400
= T 132x25 9x14 . mono . . Genoa SuperEGA BIOS 3.0+
= G 100x40 8x15 800x600 2 . . AT&T VDC600
47h = T 132x29 8x12 . mono . . Genoa 6400
= T 132x29 9x12 . mono . . Genoa SuperEGA BIOS 3.0+
= T 132x28 9x16 1188x448 16/256K . B800 WD90C
= T 132x28 9x16 . 16 . B800 Diamond Speedstar 24X
= G 100x37 8x16 800x600 16 . . AT&T VDC600
48h = T 132x32 8x12 . mono . . Genoa 6400
= T 132x32 9x11 . mono . . Genoa SuperEGA BIOS 3.0+
= G 80x50 8x8 640x400 2 . B800 AT&T 6300, AT&T VDC600
= G 80x50 8x8 640x400 2 . B800 Olivetti Quaderno
49h = T 132x44 8x8 . mono . . Genoa 6400
= T 132x44 9x8 . mono . . Genoa SuperEGA BIOS 3.0+
= G 80x30 8x16 640x480 . . . Lava Chrome II EGA
= G 80x30 8x16 640x480 . . A000 Diamond Stealth64 Video 2xx1
4Bh = G 100x37 8x16 800x600 . . A000 Diamond Stealth64 Video 2xx1
4Dh = T 120x25 . . . . . VEGA VGA
= G . . 512x480 16M . . Compaq QVision 1024/1280
= G 128x48 8x16 1024x768 . . A000 Diamond Stealth64 Video 2xx1
4Eh = T 120x43 . . . . . VEGA VGA
= T 80x60 8x8 . 16/256K . B800 Oak OTI-067/OTI-077 [8]
= G . . 640x400 16M . . Compaq QVision 1024/1280
= G 144x54 8x16 1152x864 . . A000 Diamond Stealth64 Video 2xx1
4Fh = T 132x25 . . . . . VEGA VGA
= T 132x60 . . . . . some Oak Tech VGA [8]
= G . . 640x480 16M . . Compaq QVision 1280
50h = T 80x30 8x16 . 16/256K . B800 Trident TVGA 8800/8900

```

```

= T 80x34 . . . . . Lava Chrome II EGA
= T 80x43 . . mono . . VEGA VGA
= T 132x25 9x14 . mono . . Ahead Systems EGA2001
= T 132x25 9x14 . 4 4 B800 Ahead B
= T 132x25 8x14 . 16 8 B800 OAK Technologies VGA-16
= T 132x25 8x14 . 16/256K . B800 Oak OTI-037/067/077 [8]
= T 132x25 8x14 1056x350 16 8 B800 UM587 chipset
= T 132x30 . . 16 . . MORSE VGA
= T 132x30 . . . . . Cirrus 510/520 chipset
= G 80x30 8x16 640x480 16 . . Paradise EGA-480
= G 80x30 8x16 640x480 16 . . NEL Electronics BIOS
= G 80x30 8x16 640x480 16M . . Chips&Technologies 64310
= G . . 640x480 mono??? . . Taxan 565 EGA
= G 40x25 8x8 320x200 . . . . Genoa SuperEGA BIOS 3.0+
51h = T 80x30 8x16 . . . . . Paradise EGA-480
= T 80x30 9x16 . . . . . NEL Electronics BIOS
= T 80x30 . . . . . Lava Chrome II EGA
= T 80x43 8x11 . 16/256K . B800 Trident TVGA 8800/8900
= T 132x25 . . mono . . VEGA VGA
= T 132x28 9x12 . 4 4 B800 Ahead B
= T 132x43 8x8 . 16 5 B800 OAK Technologies VGA-16
= T 132x43 8x8 . 16/256K . B800 Oak OTI-037/067/077
= T 132x43 8x8 1056x344 16 5 B800 UM587 chipset
= T 132x50 . . 16 . . MORSE VGA
= T 132x50 . . . . . Cirrus 510/520 chipset
= G 80x34 8x14 640x480 16 . . ATI EGA Wonder
= G 80x25 8x8 640x200 . . . . Genoa SuperEGA BIOS 3.0+
52h = T 80x60 . . . . . Lava Chrome II EGA
= T 80x60 8x8 . 16/256K . B800 Trident TVGA 8800/8900
= T 132x43 . . mono . . VEGA VGA
= T 132x44 9x8 . mono . . Ahead Systems EGA2001
= T 132x44 9x8 . 4 2 B800 Ahead B
= T 132x60 . . 16 . . MORSE VGA
= T 132x60 . . . . . Cirrus 510/520 chipset
= G 80x25 8x19 640x480 16 1 A000 AX VGA (Kanji&superimpose)
= G 94x29 8x14 752x410 16 . . ATI EGA Wonder
= G 100x75 8x8 800x600 16 1 A000 OAK Technologies VGA-16
= G 100x75 8x8 800x600 16 . A000 Oak OTI-037 chipset [8]
= G 100x37 8x16 800x600 16 . A000 Oak OTI-067/077 chips [8]
= G 100x75 8x8 800x600 16 . A000 UM587 chipset
= G 128x30 8x16 1024x480 16 . . NEL Electronics BIOS

```

```

53h = T 80x25 8x16 . . . . NEL Electronics BIOS
      = T 80x60 . . 16 . . MORSE VGA
      = T 80x60 . . . . Cirrus 510/520 chipset
      = T 132x25 8x14 . 16/256K . B800 Trident TVGA 8800/8900
      = T 132x43 . . . . Lava Chrome II EGA
      = G 80x25 8x19 640x480 16 1 A000 AX VGA (Kanji, no superimp.)
      = G . . 640x480 256 . . Oak VGA
      = G 80x30 8x16 640x480 256 . A000 Oak OTI-067/OTI-077 [8]
      = G 100x40 8x14 800x560 16 . . ATI EGA Wonder,ATI VIP
      = G . . . . AX PC
54h = T 132x25 . . . . Lava Chrome II EGA
      = T 132x30 8x16 . 16/256K . B800 Trident TVGA 8800/8900
      = T 132x43 8x8 . . . . Paradise EGA-480
      = T 132x43 8x8 . . . . NEL Electronics BIOS
      = T 132x43 7x9 . 16/256K . B800 Paradise VGA
      = T 132x43 8x9 . 16/256K . B800 Paradise VGA on multisync
      = T 132x43 . . . . Taxan 565 EGA
      = T 132x43 . . . . AST VGA Plus
      = T 132x43 . . . . Hewlett-Packard D1180A
      = T 132x43 7x9 . 16 . . AT&T VDC600
      = T 132x43 9x9 1188x387 16/256K . B800 WD90C
      = T 132x43 9x9 1188x387 16/256K . B800 Diamond Speedstar 24X
      = T 132x43 9x9 1188x387 16/256K . B800 Diamond Stealth 24
      = T 132x43 8x8 . . . B800 Diamond Stealth64 Video 2xx1
      = T 132x43 8x8 1056x350 16/256K . . Cirrus CL-GD5420/5422/5426
      = T 132x50 8x8 . 16 . A000 NCR 77C22 [9]
      = G 100x42 8x14 800x600 16 . A000 ATI EGA Wonder, VGA Wonder
      = G 100x42 8x14 800x600 16 . A000 ATI Ultra 8514A, ATI XL
      = G . . 800x600 256 . A000 Oak VGA
      = G 100x37 8x16 800x600 256 . A000 Oak OTI-067/077 chips [8]
55h = T 80x66 8x8 . 16/256K . A000 ATI VIP
      = T 132x25 8x14 . . . . Paradise EGA-480
      = T 132x25 8x14 . . . . NEL Electronics BIOS
      = T 132x25 7x16 . 16/256K . B800 Paradise VGA
      = T 132x25 8x16 . 16/256K . B800 Paradise VGA on multisync
      = T 132x25 . . . . Taxan 565 EGA
      = T 132x25 . . . . AST VGA Plus
      = T 132x25 . . . . Hewlett-Packard D1180A
      = T 132x25 7x16 . 16 . . AT&T VDC600
      = T 132x25 8x16 . 16 . A000 NCR 77C22 [9]
      = T 132x25 9x16 1188x400 16/256K . B800 WD90C

```

```

= T 132x25 9x16 1188x400 16/256K . B800 Diamond Speedstar 24X
= T 132x25 9x16 1188x400 16/256K . B800 Diamond Stealth 24
= T 132x25 8x16 . . . B800 Diamond Stealth64 Video 2xx1
= T 132x25 8x14 1056x350 16/256K . . Cirrus CL-GD5420/5422/5426
= T 132x43 8x11 . 16/256K . B800 Trident TVGA 8800/8900
= G 94x29 8x14 752x410 . . . Lava Chrome II EGA
= G 128x48 8x16 1024x768 16/256K . A000 ATI VGA Wonder v4+ [5]
= G . . 1024x768 16/256K . . ATI VGA Wonder Plus
= G . . 1024x768 16/256K . . ATI Ultra 8514A,ATI XL
= G 128x48 8x16 1024x768 4 . A000 Oak OTI-067/077 chips [8]
56h = T 132x43 8x8 . 3??? 2 B000 NSI Smart EGA+
= T 132x43 7x9 . 4 . B000 Paradise VGA
= T 132x43 8x9 . 4 . B000 Paradise VGA on multisync
= T 132x43 . . mono . . Taxan 565 EGA
= T 132x43 7x9 . 2 . . AT&T VDC600
= T 132x43 9x8 . . . . NEL Electronics BIOS
= T 132x50 8x8 . 4 . A000 NCR 77C22 [9]
= T 132x60 8x8 . 16/256K . B800 Trident TVGA 8800/8900
= G . . 1024x768 16 . A000 Oak VGA
= G 128x48 8x16 1024x768 16 . A000 Oak OTI-067/077 chips [8]
57h = T 132x25 8x14 . 3??? 4 B000 NSI Smart EGA+
= T 132x25 7x16 . 4 . B000 Paradise VGA
= T 132x25 8x16 . 4 . B000 Paradise VGA on multisync
= T 132x25 9x14 . . . . NEL Electronics BIOS
= T 132x25 . . mono . . Taxan 565 EGA
= T 132x25 7x16 . 2 . . AT&T VDC600
= T 132x25 9x14 . 16/256K . B800 Trident TVGA 8800/8900
= T 132x25 8x16 . 4 . A000 NCR 77C22 [9]
= G 96x48 8x16 768x1024 16 . A000 Oak OTI-067/077 chips [8]
58h = T 80x33 8x14 . 16 . B800 ATI EGA Wonder,ATI VIP
= T 80x32 9x16 . 16 . . Genoa 6400
= T 80x43 8x8 . . . . NEL Electronics BIOS
= T 132x30 9x16 . 16/256K . B800 Trident TVGA 8800/8900
= G 100x75 8x8 800x600 16/256K . A000 Paradise VGA
= G 100x75 8x8 800x600 16 . . AT&T VDC600
= G 100x75 8x8 800x600 16 . A000 NCR 77C22 [9]
= G 100x75 8x8 800x600 16 . A000 Diamond Speedstar 24X
= G 100x75 8x8 800x600 16/256K . A000 Paradise VGA, WD90C
= G . . 800x600 16 . . AST VGA Plus, Compaq VGA
= G . . 800x600 16 . . Dell VGA
= G . . 800x600 16 . . Hewlett-Packard D1180A

```

```

= G      .      .      800x600  ???  . .  ELT VGA PLUS 16
= G 100x37 8x16 800x600 16/256K .  A000 Cirrus CL-GD5420/5422/5426
= G 160x64 8x16 1280x1024 16 .  A000 Oak OTI-077 chipset [8]
59h = T 80x43 9x8 . . . . NEL Electronics BIOS
= T 80x66 8x8 . 16/256K .  A000 ATI VIP
= T 132x43 9x11 . 16/256K .  B800 Trident TVGA 8800/8900
= G 100x75 8x8 800x600 2 .  A000 Paradise VGA
= G 100x75 8x8 800x600 2 . . AT&T VDC600
= G      .      .      800x600  2  . .  AST VGA Plus, Compaq VGA
= G      .      .      800x600  2  . .  Dell VGA
= G      .      .      800x600  2  . .  Hewlett-Packard D1180A
= G 100x75 8x8 800x600 2 .  A000 NCR 77C22 [9]
= G 128x48 8x16 1024x768 256 .  A000 Oak OTI-077 chipset [8]
5Ah = T 80x60 8x8 . . . . NEL Electronics BIOS
= T 132x60 9x8 . 16/256K .  B800 Trident TVGA 8800/8900
= G 128x48 8x16 1024x768 2 .  A000 NCR 77C22 [9]
5Bh = T 80x30 8x16 . . . . B800 ATI VGA Wonder (undoc)
= G      .      .      640x350 256 . .  Genoa 6400
= G 80x25 8x16 640x400 32K .  A000 Oak OTI-067/077 chips [8]
= G      .      .      800x600 16 . .  Maxxon, SEFCO TVGA, Imtec
= G 100x75 8x8 800x600 16/256K .  A000 Trident TVGA 8800, 8900
= G      .      .      800x600  ???  . .  Vobis MVGA
= G 100x37 8x16 800x600 . . . . NEL Electronics BIOS
= G 128x48 8x16 1024x768 16 .  A000 NCR 77C22 [1,9]
5Ch = T 100x37 8x16 . . . . NEL Electronics BIOS
= G      .      .      640x400 256 . .  Logix, ATI Prism Elite
= G      .      .      640x400 256 . .  Maxxon, SEFCO TVGA, Imtec
= G 80x25 8x16 640x400 256/256K .  A000 Zymos Poach, Hi Res 512
= G 80x25 8x16 640x400 256/256K .  A000 Trident TVGA 8800/8900
= G 80x30 8x16 640x480 256 . .  Genoa 6400
= G 80x30 8x16 640x480 32K .  A000 Oak OTI-077 chipset [8]
= G 100x75 8x8 800x600 256 .  A000 NCR 77C22 [9]
= G 100x75 8x8 800x600 256/256K .  A000 WD90C
= G 100x75 8x8 800x600 256/256K .  A000 Diamond Speedstar 24X
= G 100x37 8x16 800x600 256/256K .  A000 Cirrus CL-GD5420/5422/5426
5Dh = T 100x75 8x8 . . . . NEL Electronics BIOS
= G 80x25 8x14 640x350 64K . .  STB Lightspeed ET4000/W32P
= G      .      .      640x480 256 . .  Logix, ATI Prism Elite
= G      .      .      640x480 256 . .  Maxxon, SEFCO TVGA, Imtec
= G 80x30 8x16 640x480 256/256K .  A000 Zymos Poach, Hi Res 512
= G 80x30 8x16 640x480 256/256K .  A000 Trident TVGA 8800 (512K)

```

```

= G 128x48 8x16 1024x768 16 . A000 NCR 77C22 [9]
= G 128x48 8x16 1024x768 16/256K . A000 WD90C
= G 128x48 8x16 1024x768 16 . A000 Diamond Speedstar 24X
= G 128x48 8x16 1024x768 16/256K . A000 Cirrus CL-GD5420/5422/5426
5Eh = G . . 640x400 256 . . Paradise VGA,VEGA VGA
= G . . 640x400 256 . . AST VGA Plus, NCR 77C22
= G . . 640x400 256 . . Compaq VGA, Dell VGA
= G 80x25 8x16 640x400 256 . . AT&T VDC600
= G 80x25 8x16 640x400 256 . A000 NCR 77C22 [9]
= G 80x25 8x16 640x400 256/256K . A000 WD90C
= G 80x25 8x16 640x400 256/256K . A000 Diamond Speedstar 24X
= G . . 800x600 16 . . Logix, ATI Prism Elite
= G 100x37 8x16 800x600 16 . . NEL Electronics BIOS
= G 100x75 8x8 800x600 256 . . Genoa 6400
= G 100x75 8x8 800x600 256/256K . A000 Zymos Poach, Trident 8900
= G 100x75 8x8 800x600 256/256K . A000 Hi Res 512
5Fh = G 80x25 8x16 640x400 64K . . STB Lightspeed ET4000/W32P
= G . . 640x480 256 . . Paradise VGA
= G . . 640x480 256 . . AST VGA Plus, NCR 77C22
= G . . 640x480 256 . . Compaq VGA, Dell VGA
= G . . 640x480 256 . . Hewlett-Packard D1180A
= G 80x30 8x16 640x480 256 . . AT&T VDC600 (512K)
= G 80x30 8x16 640x480 256 . A000 NCR 77C22 [9]
= G 80x30 8x16 640x480 256/256K . A000 WD90C
= G 80x30 8x16 640x480 256/256K . A000 Diamond Speedstar 24X
= G 80x30 8x16 640x480 256/256K . A000 Cirrus CL-GD5420/5422/5426
= G . . 1024x768 16 . . Logix, ATI Prism Elite
= G . . 1024x768 16 . . Maxxon, Imtec
= G 128x48 8x16 1024x768 16 . . Genoa 6400
= G 128x48 8x16 1024x768 16/256K . A000 Zymos Poach, Hi Res 512
= G 128x48 8x16 1024x768 16/256K . A000 Trident TVGA 88/8900 512K
60h = T 132x25 8x14 . 16/64 8 B800 Quadram Ultra VGA
= T 132x25 8x14 . 16 . . Genoa 6400
= T 132x25 8x14 . 16 . . Genoa SuperEGA BIOS 3.0+
= T 132x25 . . . . . Cirrus 5320 chipset
= T 132x25 8x16 1056x400 16 . B800 Chips&Technologies chipset
= G 80x??? . ???x400 . . . Corona/Cordata BIOS 4.10+
= G 80x25 8x16 640x400 256 1 A000 Ahead A, Ahead B
= G . . 752x410 . . . VEGA VGA
= G . . 752x410 16 . . Tatung VGA
= G . . 752x410 16 . . Video7 V-RAM VGA

```

```

= G 128x48 8x16 1024x768 4/256K . A000 Trident TVGA 8900
= G 128x48 8x16 1024x768 256/256K . A000 WD90C
= G 128x48 8x16 1024x768 256/256K . A000 Diamond Speedstar 24X
= G 128x48 8x16 1024x768 256/256K . A000 Cirrus CL-GD5420/5422/5426
= G 144x54 8x16 1152x864 . . A000 Diamond Stealth64 Video 2xx1
61h = T 132x29 8x12 . 16/64 8 B800 Quadram Ultra VGA
= T 132x29 8x8 . 16 . . Genoa 6400
= T 132x29 8x8 . 16 . . Genoa SuperEGA BIOS 3.0+
= T 132x50 . . . . . Cirrus 5320 chipset
= T 132x50 8x8 1056x400 16 . B800 Chips&Technologies chipset
= T 132x50 8x16 1056x800 16 . B800 Chips&Technologies 64310
= G . . ???x400 . . . Corona/Cordata BIOS 4.10+
= G 80x25 8x16 640x400 256 . A000 ATI VGA Wonder,VGA Wonder+
= G 80x25 8x16 640x400 256 . A000 ATI Ultra 8514A,ATI XL
= G 80x25 8x16 640x400 . . A000 Diamond Stealth64 Video 2xx1
= G 80x30 8x16 640x480 256 1 A000 Ahead A, Ahead B (512K)
= G . . 720x540 . . . VEGA VGA
= G . . 720x540 16 . . Tatung VGA
= G . . 720x540 16 . . Video7 V-RAM VGA
= G 96x64 8x16 768x1024 16/256K . A000 Trident TVGA 88/8900 512K
= G 128x48 8x16 1024x768 256 . A000 NCR 77C22 [1,9]
= G 144x54 8x16 1152x864 . . A000 Diamond Stealth64 Video 2xx1
62h = T 132x32 8x11 . 16/64 6 B800 Quadram Ultra VGA
= T 132x32 8x12 . 16 . . Genoa 6400
= T 132x32 8x11 . 16 . . Genoa SuperEGA BIOS 3.0+
= T 132x43 8x8 1056x344 16 . B800 C&T 82C450 BIOS
= G . . 640x450 16 . . Cirrus 510/520 chipset
= G 80x30 8x16 640x480 256 . A000 ATI VGA Wonder,VGA Wonder+
= G 80x30 8x16 640x480 256 . A000 ATI Ultra 8514A,ATI XL
= G 80x30 8x16 640x480 32K . A000 WD90C
= G 80x30 8x16 640x480 32K . A000 Diamond Speedstar 24X
= G . . 800x600 . . . VEGA VGA
= G . . 800x600 16 . . Tatung VGA
= G . . 800x600 16 . . Video7 V-RAM VGA
= G 100x75 8x8 800x600 256 1 A000 Ahead A, Ahead B (512K)
= G 128x48 8x16 1024x768 256/256K . A000 Trident TVGA 8900, Zymos
= G 128x48 8x16 1024x768 256 . A000 NCR 77C22 [9]
63h = T 132x44 8x8 . 16/64 5 B800 Quadram Ultra VGA
= T 132x44 8x8 . 16 . . Genoa 6400
= T 132x44 8x8 . 16 . . Genoa SuperEGA BIOS 3.0+
= G . . 720x540 16 . . MORSE VGA

```



```

= G      .      .      720x540  16      . .  Cirrus 510/520 chipset
= G 100x42 8x14  800x600  256      .   A000 ATI VGA Wonder,VGA Wonder+
= G 100x42 8x14  800x600  256      .   A000 ATI Ultra 8514A,ATI XL
= G      .      .      800x600  32K      .   A000 WD90C
= G      .      .      800x600  32K      .   A000 Diamond Speedstar 24X
= G 128x48 7x16 1024x768  256      1   A000 Ahead B (1MB)
= G      .      .      1024x768   2      . .  Video7 V-RAM VGA
64h = T 132x60 8x8      .  16      . .  Genoa 6400
= T  80x43 8x8      528x344  16      .   B800 C&T 82C450 BIOS
= G      .      .      640x480  64K      .   A000 Cirrus CL-GD 5422/5426
= G      .      .      800x600   16      . .  MORSE VGA
= G      .      .      800x600   16      . .  Cirrus 510/520 chipset
= G      .      .      800x600   ???     . .  SAMPO-Mira VGA
= G      .      .      1024x768   4      . .  Video7 V-RAM VGA
= G 128x48 8x16 1024x768  256      .   A000 ATI VGA Wonder Plus,ATI XL
= G 160x64 8x16 1280x1024 16/256K .   A000 WD90C [1]
= G 160x64 8x16 1280x1024 16/256K .   A000 Diamond Speedstar 24X [1]
65h = T  80x50 8x8      528x400  16      .   B800 C&T 82C450 BIOS
= G      .      .      800x600  64K      .   A000 Cirrus CL-GD 5422/5426
= G      .      .      1024x768  16      . .  Video7 V-RAM VGA
= G 128x48 8x16 1024x768  16      .   A000 ATI VGA Wonder
66h = T  80x50 8x8      640x400  16/256K .   B800 WD90C
= T  80x50 8x8      .  16      .   B800 Diamond Speedstar 24X
= G      .      .      640x400  256      . .  Tatung VGA
= G      .      .      640x400  256      . .  Video7 V-RAM VGA
= G      .      .      640x480  32K      .   A000 Cirrus CL-GD 5422/5426
67h = T  80x43 8x8      640x344  16/256K .   B800 WD90C
= T  80x43 8x8      .  16      .   B800 Diamond Speedstar 24X
= G      .      .      640x480  256      . .  Video7 V-RAM VGA
= G      .      .      800x600  32K      .   A000 Cirrus CL-GD 5422/5426
= G 128x48 8x16 1024x768   4      .   A000 ATI VGA Wonder
= G 160x64 8x16 1280x1024  16      .   A000 NCR 77C22 [1,9]
68h = G  80x25 8x16  640x400      . .   A000 Diamond Stealth64 Video 2xx1
69h = T 132x50 8x8      1056x400 16/256K .   B800 WD90C
= T 132x50 8x8      .  16      .   B800 Diamond Speedstar 24X
= G  80x30 8x16  640x480      . .   A000 Diamond Stealth64 Video 2xx1
= G      .      .      720x540  256      .   A000 Video7 V-RAM VGA
6Ah = G      .      .      800x600   16      .   A000 VESA standard interface
= G 100x75 8x8      800x600   16      .   A000 Genoa 6400
= G 100x75 8x8      800x600   16      .   A000 Diamond Speedstar 24X
= G      .      .      800x600   16      .   A000 Ahead A

```

```

= G 100x75 8x8 800x600 16 1 A000 Ahead B (VESA) [see 71h]
= G . . 800x600 16 . . Zymos Poach, Hi Res 512
= G . . 800x600 16 . . Epson LT-386SX in CRT Mode
= G . . 800x600 16 . . Compuadd 316SL in CRT Mode
= G 100x37 8x16 800x600 16/256K . A000 Cirrus CL-GD5420/5422/5426
= G 100x37 8x16 800x600 16 . A000 Diamond Stealth64 Video 2xx1
= G 100x42 8x14 800x600 . . A000 ATI VGA Wonder (undoc)
= G . . 800x600 16 . A000 Chips&Technologies chipset
= G 160x64 8x16 1280x1024 256 . A000 NCR 77C22 [1,9]
6Bh = T 100x37 8x16 . 16 . . Genoa 6400
= T 100x37 8x16 . . . . NEL Electronics BIOS
= G 100x37 8x16 800x600 . . A000 Diamond Stealth64 Video 2xx1
6Ch = G 80x30 8x16 640x480 16M . A000 Trident 8900CL/BIOS C04
= G 100x75 8x8 800x600 256 . . Genoa 6400
= G 128x48 8x16 1024x768 2 . A000 Diamond Stealth64 Video 2xx1
= G 160x60 8x16 1280x960 16/256K . A000 WD90C [1]
= G 160x60 8x16 1280x960 16/256K . A000 Diamond Speedstar 24X [1]
= G 160x64 8x16 1280x1024 16/256K . A000 Cirrus CL-GD 5422/5426 [1]
6Dh = G 80x25 8x14 640x350 64K . A000 STB Lightspeed ET4000/W32P
= G 128x48 8x16 1024x768 . . A000 Diamond Stealth64 Video 2xx1
= G 160x64 8x16 1280x1024 256/256K . A000 Cirrus CL-GD 5422/5426 [1]
6Eh = G 40x25 8x8 320x200 64K . A000 Cirrus CL-GD 5422/5426
= G 160x64 8x16 1280x1024 2 . A000 Diamond Stealth64 Video 2xx1
6Fh = G 40x25 8x8 320x200 16M . A000 Cirrus CL-GD 5422/5426
= G 160x64 8x16 1280x1024 . . A000 Diamond Stealth64 Video 2xx1
70h = extended mode set (see AX=0070h) . Everex Micro Enhancer EGA
= T 40x25 8x8 . 16 8 B800 Quadram (CGA double scan)
= T 40x25 8x8 (CGA dblscan) . . Genoa SuperEGA BIOS 3.0+
= G . . 360x480 256 . . Cirrus 510/520/5320 chips
= G 90x28 8x14 720x392 16 1 A000 Ahead B
= G 80x30 8x16 640x480 . . A000 Diamond Stealth64 Video 2xx1
= G 100x38 8x16 800x600 16 . A000 C&T chipset, Cardinal
= G . . 1024x480 256 . A000 Trident 8900C BIOS C3.0
71h = T 80x25 8x8 . 16 8 B800 Quadram (CGA double scan)
= T 80x25 8x8 (CGA dblscan) . . Genoa SuperEGA BIOS 3.0+
= G . . 528x400 256 . . Cirrus 510/520 chipset
= G 80x30 8x16 640x480 16M . A000 Cirrus CL-GD 5422/5426
= G 80x30 8x16 640x480 . . A000 Diamond Stealth64 Video 2xx1
= G 100x35 8x16 800x600 16/64 . A000 NSI Smart EGA+
= G 100x75 8x8 800x600 16 1 A000 Ahead B (same as 6Ah)
= G . . 960x720 16 . . C&T chipset, Cardinal

```

```

= G      .      .      1024x480  256      .      A000 Trident 8900C BIOS C3.0
72h = T  80x60  8x8      .      16      .      B800 Quadram Ultra VGA
= T  80x60  8x8      .      16      .      B800 Genoa 6400
= T  80x60  8x8      .      16      .      B800 Genoa SuperEGA BIOS 3.0+
= G      .      .      528x480  256      .      .      Cirrus 510/520 chipset
= G  80x25  8x19  640x480  16      1      A000 DOS/V w/ any VGA
= G  80x30  8x16  640x480      .      .      A000 Diamond Stealth64 Video 2xx1
= G      .      .      640x480  32K      .      A000 ATI
= G      .      .      640x480  16M      .      A000 WD90C
= G      .      .      640x480  16M      .      A000 Diamond Speedstar 24X
= G      .      .      1024x768  16      .      .      C&T chipset, Cardinal
= G 128x48  8x16 1024x768i  16      .      A000 C&T 82C450 BIOS
= G 128x48  8x16 1024x768  16      .      A000 C&T 65530 BIOS (multisync)
73h = G  80x60  8x8      640x480  16      .      A000 Quadram Ultra VGA
= G  80x60  8x8      640x480  16      .      .      Genoa 6400
= G  80x60  8x8      640x480  16      .      .      Genoa SuperEGA BIOS 3.0+
= G 100x37  8x16  800x600      .      .      A000 Diamond Stealth64 Video 2xx1
= T  80x25  8x19  640x475  16      1      none DOS/V, emulated in VGA graph
74h = T  80x66  8x8      .      16      .      B800 Quadram Ultra VGA
= T  80x66  8x8      .      16      .      B800 Genoa 6400
= T  80x66  8x8      .      16      .      B800 Genoa SuperEGA BIOS 3.0+
= G      .      .      640x400    2      .      B800 Toshiba 3100 AT&T mode
= G  80x30  8x16  640x480  32K      .      A000 Trident 8900C/BIOS C03
= G 100x37  8x16  800x600      .      .      A000 Diamond Stealth64 Video 2xx1
= G 128x48  8x16 1024x768  16      1      A000 Ahead A, Ahead B (512K)
= G      .      .      1024x768  64K      .      A000 Cirrus CL-GD 5422/5426 [1]
75h = G  80x30  8x16  640x480  64K      .      A000 Trident 8900C/BIOS C03
= G  80x66      .      640x528  16???      .      A000 Quadram Ultra VGA
= G  80x66      .      640x528  16      .      .      Genoa SuperEGA BIOS 3.0+
= G 100x37  8x16  800x600      .      .      A000 Diamond Stealth64 Video 2xx1
= G 128x48  8x16 1024x768  4      1      A000 Ahead B
= G 128x48  8x16 1024x768  16      .      A000 Chips&Technologies 64310
76h = T  94x29  8x14      .      16      .      B800 Quadram Ultra VGA
= T  94x29  8x14      .      .      .      .      Genoa SuperEGA BIOS 3.0+
= G 100x75  8x8      800x600  32K      .      A000 Trident 8900C/BIOS C03
= G 128x48  8x16 1024x768  2      1      A000 Ahead B
= G 128x48  8x16 1024x768      .      .      A000 Diamond Stealth64 Video 2xx1
= G 160x64  8x16 1280x1024  16      .      A000 Chips&Technologies 64310 [1]
77h = G  94x29      .      752x410  16???      .      A000 Quadram Ultra VGA
= G  94x29      .      752x410  16      .      .      Genoa SuperEGA BIOS 3.0+
= G 100x75  8x8      800x600  64K      .      A000 Trident 8900C/BIOS C03

```

```

= G 128x48 8x16 1024x768 . . A000 Diamond Stealth64 Video 2xx1
78h = T 100x37 8x16 . 16 . . Genoa 6400
= T 100x75 8x8 . 16 . B800 Quadram Ultra VGA
= T 100x75 8x8 . . . . Genoa SuperEGA BIOS 3.0+
= G . . 640x400 256 . . STB VGA/EM-16 Plus
= G 80x25 8x16 640x400 256 . . Cardinal, C&T chipset
= G . . 640x400 256 . . Cirrus 5320 chipset
= G 80x25 8x16 640x400 256 . A000 Chips&Technologies 64310
79h = G 80x30 8x16 640x480 256 . . Cardinal, C&T chipset
= G 80x30 8x16 640x480 256 . A000 Chips&Technologies 64310
= G 100x75 . 800x600 16??? . A000 Quadram Ultra VGA
= G 100x75 8x8 800x600 16 . . Genoa SuperEGA BIOS 3.0+
= G 100x75 8x8 800x600 16 . . Genoa 6400
7Ah = T 114x60 8x8 . 16 . B800 Quadram Ultra VGA
= T 114x60 8x8 . . . . Genoa SuperEGA BIOS 3.0+
= G . . 720x540 256 . . C&T chipset, Cardinal
7Bh = G . . 800x600 256 . . C&T chipset, Cardinal
= G 114x60 . 912x480 16??? . A000 Quadram Ultra VGA
= G . . 912x480 16 . . Genoa SuperEGA BIOS 3.0+
7Ch = G . . 512x512 16 . . Genoa
= G 100x37 8x16 800x600 256 . . C&T 82C453/F65530 chipsets
= G 100x37 8x16 800x600 256 . A000 Chips&Technologies 64310
= G 200x75 8x16 1600x1200 . [16] . A000 Diamond Stealth64 Video 2xx1
7Dh = G 64x32 8x16 512x512 256 . . Genoa
7Eh = special mode set (see AX=007Eh) . Paradise VGA, AT&T VDC600
= G 80x25 8x16 640x400 256 . . Genoa 6400
= G . . 1024x768 256 . . C&T 82C453 chipset
= G 128x48 8x16 1024x768 256 . A000 Chips&Technologies 64310
= G 90x43 . . mono . B000 HERCULES.COM on HGC [14]
7Fh = special function set (see AX=007Fh/BH=00h) Paradise VGA, AT&T VDC600
= G 128x48 8x16 1024x768 4 . . Genoa 6400
= G 90x29 . . mono . B000 HERCULES.COM on HGC [14]
82h = T 80x25 . . B&W . . AT&T VDC overlay mode [6]
83h = T 80x25 . . . . AT&T VDC overlay mode [6]
86h = G . . 640x200 B&W . . AT&T VDC overlay mode [6]
88h = G 90x43 8x8 720x348 mono . B000 Hercules + MSHERC.COM
C0h = G . . 640x400 2/prog palette . AT&T VDC overlay mode [6]
= G . . 640x400 2/prog palette . Olivetti Quaderno overlay
C4h = disable output . . . . AT&T VDC overlay mode [6]
C8h = G 80x50 8x8 640x400 2 . B800 Olivetti Quaderno overlay
D0h = G . . 640x400 2 . B800 DEC VAXmate AT&T mode

```

Notes:

- [1] interlaced only
- [2] for ATI EGA Wonder, mode 08h is only valid if SMS.COM is loaded resident.
SMS maps mode 08h to mode 27h if the byte at location 0040:0063 is 0B4h,
otherwise to mode 23h, thus selecting the appropriate (monochrome or
color) 132x25 character mode.
for ATI VGA Wonder, mode 08h is the same, and only valid if VCONFIG loaded
resident
- [3] early XGA boards support 132-column text but do not have this BIOS mode
- [4] DESQview intercepts calls to change into these two modes (21h is page 0,
22h is page 1) even if there is no Hercules graphics board installed
- [5] ATI BIOS v4-1.00 has a text-scrolling bug in this mode
- [6] for AT&T VDC overlay modes, BL contains the DEB mode, which may be 06h,
40h, or 44h
- [7] BIOS text support is broken in this undocumented mode; scrolling moves
only about 1/3 of the screen (and does even that portion incorrectly),
while screen clears only clear about 3/4.
- [8] The Oak OTI-037/067/077 modes are present in the Oak VGA BIOS, which OEMs
may choose to use only partially or not at all; thus, not all Oak boards
support all "Oak" modes listed here
- [9] this card uses the full 128K A000h-BFFFh range for the video buffer,
precluding the use of a monochrome adapter in the same system
- [10] mode 17h supported by Tseng ET4000 BIOS 8.01X dated 1990/09/14, but not
v8.01X dated 1992/02/28; mode 21h supported by 1992/02/28 version but not
1990/09/14 version
- [11] HERKULES simulates a 90x45 text mode in Hercules graphics mode; the
installation check for HERKULES.COM is the signature "Herkules" two
bytes beyond the INT 10 handler
- [12] The Realtek RTVGA BIOS v3.C10 crashes when attempting to switch into
modes 21h or 27h; this version of the BIOS also sets the BIOS data area
incorrectly for extended text modes, resulting in scrolling after only
24 lines (the VMODE.EXE utility does set the data area correctly)
- [13] The Tandy 1000SL/TL BIOS does not actually support this mode
- [14] HERCULES.COM is a graphics-mode BIOS extension for Hercules-compatible
graphics cards by Soft Warehouse, Inc. Its installation check is to
test whether the word preceding the INT 10 handler is 4137h.
- [15] The Hercules-graphics video modes for HERCBIOS (shareware by Dave
Tutelman) may be changed by a command-line switch; the 90x43
character-cell mode's number is always one higher than the 90x29 mode
(whose default is mode 08h)
- [16] Stealth64 Video 2001-series BIOS v1.03 reports 76 lines for mode 7Ch,

resulting in incorrect scrolling for TTY output (scrolling occurs only
after the end of the 76th line, which is not displayed)

[17] For 43-line text on EGA or 43/50-line text on VGA, you must load an 8x8
font using AX=1102h after switching to mode 3; VGA may also require
using INT 10/AH=12h/BL=30h

SeeAlso: #00011,#00083,#00191

Index: video modes

Index: installation check;HERKULES|installation check;HERCULES.COM

-----V-100070-----

INT 10 - VIDEO - Everex Micro Enhancer EGA/Viewpoint VGA - EXTENDED MODE SET

AX = 0070h

BL = mode (see #00011)

Desc: specify a proprietary display mode on the Everex Micro Enhancer or
Viewpoint video adapters

SeeAlso: AH=00h,AX=6F05h,AX=7000h/BX=0004h,AH=FFh"GO32"

(Table 00011)

Values for Everex video mode:

	text/ grph	text resol	pixel box	pixel resolution	colors	disply pages	scrn addr	monitor	adapter
00h = G	.	.	640x480	16	. .	multisync	EGA,VGA		
01h = G	.	.	752x410	16	. .	multisync	EGA,VGA		
02h = G	.	.	800x600	16	. .	multisync	EGA,VGA		
03h = T	80x34	multisync	EGA,VGA		
04h = T	80x60	multisync	EGA,VGA		
05h = T	94x29	multisync	EGA only		
06h = T	94x51	multisync	EGA only		
07h = T	100x43	8x14	.	16	. .	.	VGA only		
08h = T	100x75	8x8	.	16	. .	.	VGA only		
09h = T	80x44	EGA	EGA only		
0Ah = T	132x25	EGA	EGA,VGA		
0Bh = T	132x44	EGA	EGA,VGA		
0Ch = T	132x25	CGA	EGA only		
0Dh = T	80x44	mono	EGA only		
0Eh = T	132x25	mono			
0Fh = T	132x44	mono			
10h =	reserved								
11h = G	.	.	1280x350	4	. .	.	EGA only		
12h = G	.	.	1280x600	4	. .	.	EGA only		
13h = G	.	.	640x350	256	. .	.	EGA,EV673		
14h = G	.	.	640x400	256	. .	.			

```

15h = G      .      .      512x480  256  . . .
16h = T  80x30  8x16  . 256  . . .      VGA only
18h = T 100x27  8x16  . 16  . . .      VGA only
20h = G      .      .      1024x768  16  . . .      Everex 629,678
      .      .      .      .      .      .      Everex EV-673
21h = T 160x64  8x16 1280x1024 16  . . .      1MB VGA only
30h = G      .      .      640x480  256  . . .      Everex 629,678
      .      .      .      .      .      .      Everex EV-673
31h = G      .      .      800x600  256  . . .      Everex 629,678
      .      .      .      .      .      .      Everex EV-673
32h = G 128x48  8x16 1024x768  256  . . .      1MB VGA only
40h = T 132x30  8x16  . 16  . . .      VGA only
50h = T 132x32  8x16  . mono . . .      VGA only
62h = G  40x25  8x8   320x200  32K  . . .      Vwpt TC (EV629)
70h = G  64x30  8x16  512x480  32K  . . .      Viewpoint TC
71h = G  80x30  8x16  640x480  32K  . . .      Viewpoint TC
76h = G  64x30  8x16  512x480  16M  . . .      Viewpoint TC
77h = G  80x30  8x16  640x480  16M  . . .      Viewpoint TC

```

SeeAlso: #00010,#00191

Index: video modes;Everex

-----V-10007E-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - SET SPECIAL MODE

AX = 007Eh

BX = horizontal dimension of the mode desired

CX = vertical dimension of the mode desired

(both BX/CX in pixels for graphics modes, rows for alpha modes)

DX = number of colors of the mode desired (0000h for monochrome modes)

Return: BH = 7Eh if successful (Paradise VGA)

AL = 7Eh if successful (AT&T VDC600)

Desc: specify a display mode by resolution rather than mode number

SeeAlso: AH=00h,AX=0070h,AX=007Fh/BH=00h,AX=6F05h,AH=FFh"GO32"

Index: video modes;Paradise|video modes;AT&T

-----V-10007FBH00-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - SET VGA OPERATION

AX = 007Fh

BH = 00h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)

00h invalid function

7Fh successful

SeeAlso: AX=007Fh/BH=01h,AX=007Fh/BH=02h

-----V-10007FBH01-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - SET NON-VGA OPERATION

AX = 007Fh

BH = 01h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)

00h invalid function

7Fh successful

Note: color modes (0,1,2,3,4,5,6) will set non-VGA CGA operation, monochrome
mode 7 will set non-VGA MDA/Hercules operation

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

-----V-10007FBH02-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - QUERY MODE STATUS

AX = 007Fh

BH = 02h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)

00h invalid function

7Fh successful

---if successful---

BL = 00h if operating in VGA mode, 01h if non-VGA mode

CH = total video RAM size in 64k byte units

CL = video RAM used by the current mode

Note: under Novell DOS 7 TaskMgr Multitasker, this call always returns 0K
video memory in CX

SeeAlso: AX=007Fh/BH=00h,AX=007Fh/BH=01h

-----V-10007FBH03-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - LOCK CURRENT MODE

AX = 007Fh

BH = 03h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)

00h invalid function

7Fh successful

Desc: allows current mode (VGA or non-VGA) to survive re-boot

SeeAlso: AX=007Eh,AX=007Fh/BH=04h,AX=007Fh/BH=06h,AH=FFh"Oak"

-----V-10007FBH04-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - ENTER MDA EMULATION MODE

AX = 007Fh

BH = 04h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)
 00h invalid function
 7Fh successful

SeeAlso: AX=007Eh,AX=007Fh/BH=03h,AX=007Fh/BH=05h,AH=FFh"Oak"

-----V-10007FBH05-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - ENTER CGA EMULATION MODE

AX = 007Fh
 BH = 05h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)
 00h invalid function
 7Fh successful

SeeAlso: AX=007Eh,AX=007Fh/BH=03h,AX=007Fh/BH=04h,AH=FFh"Oak"

-----V-10007FBH06-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - ENTER MONOCHROME VGA MODE

AX = 007Fh
 BH = 06h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)
 00h invalid function
 7Fh successful

Note: this function also switches to video mode 7

SeeAlso: AX=007Eh,AX=007Fh/BH=03h,AX=007Fh/BH=07h,AH=FFh"Oak"

-----V-10007FBH07-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - ENTER COLOR VGA MODE

AX = 007Fh
 BH = 07h

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)
 00h invalid function
 7Fh successful

Note: this function also switches to video mode 3

SeeAlso: AX=007Eh,AX=007Fh/BH=03h,AX=007Fh/BH=06h,AH=FFh"Oak"

-----V-10007F-----

INT 10 - VIDEO - Paradise VGA, AT&T VDC600 - EXTENDED DATA REGISTER ACCESS

AX = 007Fh
 BH = function
 0Ah,0Bh,0Ch,0Dh,0Eh,0Fh WRITE PARADISE REGISTERS 0,1,2,3,4,5
 (port 03CEh indices 0Ah,0Bh,0Ch,0Dh,0Eh,0Fh)
 BL = value to set in the paradise register
 1Ah,1Bh,1Ch,1Dh,1Eh,1Fh READ PARADISE REGISTERS 0,1,2,3,4,5

(port 03CEh indices 0Ah,0Bh,0Ch,0Dh,0Eh,0Fh)

Return: BL = value of the paradise register

BH = 7Fh if successful

29h-2Fh ??? (Paradise VGA for Dell, 1991/07/24)

Return: AL = 7Fh if successful (AT&T VDC600)

BH = status (Paradise/Dell)

00h invalid function

7Fh successful

SeeAlso: AX=007Eh,AX=007Fh/BH=03h,AX=007Fh/BH=05h,AH=FFh"Oak"

-----V-10007FBH60-----

INT 10 - Paradise VGA - ???

AX = 007Fh

BH = 60h

BL = ??? (00h-13h)

???

Return: BH = status (00h failed or not supported, 7Fh successful)

???

Note: this function is supported by the 1993/5/14 ROM for the Dell 486D

SeeAlso: AX=007Fh/BH=61h

-----V-10007FBH61-----

INT 10 - Paradise VGA - ???

AX = 007Fh

BH = 61h

ES:DI -> buffer for list of ??? (see #00012)

Return: BH = status (00h failed or not supported, 7Fh successful)

???

Note: this function is supported by the 1993/5/14 ROM for the Dell 486D

SeeAlso: AX=007Fh/BH=60h

Format of Paradise VGA list entry:

Offset Size Description (Table 00012)

00h BYTE ??? or 00h if end of list

01h BYTE ???

02h WORD ???

04h WORD ???

06h WORD ???

-----V-10007FBXA500-----

INT 10 - Paradise VGA - ???

AX = 007Fh

BX = A500h

???

Return: BH = status (00h failed or not supported, 7Fh successful)

???

Note: this function is supported by the 1993/5/14 ROM for the Dell 486D

SeeAlso: AX=007Fh/BH=A6h

-----V-10007FBHA5-----

INT 10 - Paradise VGA - ???

AX = 007Fh

BH = A5h

BL = ???

bits 3-0: ??? (0-4)

bits 5,4: ???

???

Return: BH = status (00h failed or not supported, 7Fh successful)

???

Note: this function is supported by the 1993/5/14 ROM for the Dell 486D

-----V-10007FBHA6-----

INT 10 - Paradise VGA - ???

AX = 007Fh

BH = A6h

???

Return: BH = status (00h failed or not supported, 7Fh successful)

???

Note: this function is supported by the 1993/5/14 ROM for the Dell 486D

SeeAlso: AX=007Fh/BH=A5h

-----V-1001-----

INT 10 - VIDEO - SET TEXT-MODE CURSOR SHAPE

AH = 01h

CH = cursor start and options (see #00013)

CL = bottom scan line containing cursor (bits 0-4)

Return: nothing

Desc: specify the starting and ending scan lines to be occupied by the hardware cursor in text modes

Notes: buggy on EGA systems--BIOS remaps cursor shape in 43 line modes, but returns unmapped cursor shape

UltraVision scales size to the current font height by assuming 14-line monochrome and 8-line color fonts; this call is not valid if cursor emulation has been disabled

applications which wish to change the cursor by programming the hardware directly on EGA or above should call INT 10/AX=1130h or read 0040h:0085h first to determine the current font height

on some adapters, setting the end line greater than the number of lines

in the font will result in the cursor extending to the top of the
next character cell on the right

BUG: AMI 386 BIOS and AST Premier 386 BIOS will lock up the system if AL
is not equal to the current video mode

SeeAlso: AH=03h,AX=CD05h,AH=12h/BL=34h,#03885

Bitfields for cursor start and options:

Bit(s) Description (Table 00013)

7 should be zero

6,5 cursor blink

(00=normal, 01=invisible, 10=erratic, 11=slow)

(00=normal, other=invisible on EGA/VGA)

4-0 topmost scan line containing cursor

-----V-1002-----

INT 10 - VIDEO - SET CURSOR POSITION

AH = 02h

BH = page number

0-3 in modes 2&3

0-7 in modes 0&1

0 in graphics modes

DH = row (00h is top)

DL = column (00h is left)

Return: nothing

SeeAlso: AH=03h,AH=05h,INT 60/DI=030Bh,MEM 0040h:0050h

-----V-1003-----

INT 10 - VIDEO - GET CURSOR POSITION AND SIZE

AH = 03h

BH = page number

0-3 in modes 2&3

0-7 in modes 0&1

0 in graphics modes

Return: AX = 0000h (Phoenix BIOS)

CH = start scan line

CL = end scan line

DH = row (00h is top)

DL = column (00h is left)

Notes: a separate cursor is maintained for each of up to 8 display pages
many ROM BIOSes incorrectly return the default size for a color display
(start 06h, end 07h) when a monochrome display is attached

With PhysTechSoft's PTS ROM-DOS the BH value is ignored on entry.

SeeAlso: AH=01h,AH=02h,AH=12h/BL=34h,MEM 0040h:0050h,MEM 0040h:0060h

-----V-1004-----

INT 10 - VIDEO - READ LIGHT PEN POSITION (except VGA)

AH = 04h

Return: AH = light pen trigger flag

00h not down/triggered

01h down/triggered

DH,DL = row,column of character light pen is on

CH = pixel row (graphics modes 04h-06h)

CX = pixel row (graphics modes with >200 rows)

BX = pixel column

Desc: determine the current position and status of the light pen (if present)

Notes: on a CGA, returned column numbers are always multiples of 2 (320-column modes) or 4 (640-column modes)

returned row numbers are only accurate to two lines

-----V-1004-----

INT 10 - HUNTER 16 - GET CURSOR ADDRESS

AH = 04h

BH = page

Return: DH = row (0..24)

DL = column (0..79)

CH = cursor pixel Y-address (0..199)

CL = cursor pixel X-address (0..639)

Notes: the Husky Hunter 16 is an 8088-based ruggedized laptop. Other family members are the Husky Hunter, Husky Hunter 16/80, and Husky Hawk.

pixel coordinates are for the lower left corner of the character cell containing the cursor

SeeAlso: AH=60h"HUNTER"

-----V-1005-----

INT 10 - VIDEO - SELECT ACTIVE DISPLAY PAGE

AH = 05h

AL = new page number (00h to number of pages - 1) (see #00010)

Return: nothing

Desc: specify which of possibly multiple display pages will be visible

Note: to determine whether the requested page actually exists, use AH=0Fh to query the current page after making this call

SeeAlso: AH=0Fh,AH=43h,AH=45h,MEM 0040h:0062h,MEM 0040h:004Eh

-----V-100500-----

INT 10 - VIDEO - Corona/Cordata BIOS v4.10+ - SET GRAPHICS BITMAP BUFFER

AX = 0500h

BX = segment of buffer

Return: nothing

Desc: set the address of graphics bitmap buffer for video modes 60h and 61h

SeeAlso: AX=050Fh"Corona"

-----V-10050F-----

INT 10 - VIDEO - Corona/Cordata BIOS v4.10+ - GET GRAPHICS BITMAP BUFFER

AX = 050Fh

Return: DX = segment of graphics bitmap buffer

Desc: get the address of graphics bitmap buffer for video modes 60h and 61h

SeeAlso: AX=0500h"Corona"

-----V-100580-----

INT 10 - VIDEO - PCjr, Tandy 1000 - GET CRT/CPU PAGE REGISTERS

AX = 0580h

Return: BH = CRT page register

BL = CPU page register

Notes: the CPU page determines which 16K block of the first 128K of physical
memory will be mapped at B800h by the hardware

the CRT page determines the start address of the memory used by the
video controller

SeeAlso: AH=05h"SET CRT/CPU"

-----V-1005-----

INT 10 - VIDEO - PCjr, Tandy 1000 - SET CRT/CPU PAGE REGISTERS

AH = 05h

AL = subfunction

81h set CPU page register

BL = CPU page

82h set CRT page register

BH = CRT page

83h set both CPU and CRT page registers

BL = CPU page

BH = CRT page

Return: nothing

Notes: the CPU page determines which 16K block of the first 128K of physical
memory will be mapped at B800h by the hardware

the CRT page determines the start address of the memory used by the
video controller

SeeAlso: AX=0580h

-----V-1006-----

INT 10 - VIDEO - SCROLL UP WINDOW

AH = 06h

AL = number of lines by which to scroll up (00h = clear entire window)

BH = attribute used to write blank lines at bottom of window

CH,CL = row,column of window's upper left corner
DH,DL = row,column of window's lower right corner

Return: nothing

Note: affects only the currently active page (see AH=05h)

BUGS: some implementations (including the original IBM PC) have a bug which destroys BP

the Trident TVGA8900CL (BIOS dated 1992/9/8) clears DS to 0000h when scrolling in an SVGA mode (800x600 or higher)

SeeAlso: AH=07h,AH=12h"Tandy 2000",AH=72h,AH=73h,AX=7F07h,INT 50/AX=0014h

-----V-1007-----

INT 10 - VIDEO - SCROLL DOWN WINDOW

AH = 07h

AL = number of lines by which to scroll down (00h=clear entire window)

BH = attribute used to write blank lines at top of window

CH,CL = row,column of window's upper left corner

DH,DL = row,column of window's lower right corner

Return: nothing

Note: affects only the currently active page (see AH=05h)

BUGS: some implementations (including the original IBM PC) have a bug which destroys BP

the Trident TVGA8900CL (BIOS dated 1992/9/8) clears DS to 0000h when scrolling in an SVGA mode (800x600 or higher)

SeeAlso: AH=06h,AH=12h"Tandy 2000",AH=72h,AH=73h,INT 50/AX=0014h

-----V-1008-----

INT 10 - VIDEO - READ CHARACTER AND ATTRIBUTE AT CURSOR POSITION

AH = 08h

BH = page number (00h to number of pages - 1) (see #00010)

Return: AH = character's attribute (text mode only) (see #00014)

AH = character's color (Tandy 2000 graphics mode only)

AL = character

Notes: for monochrome displays, a foreground of 1 with background 0 is underlined

the blink bit may be reprogrammed to enable intense background colors using AX=1003h or by programming the CRT controller

the foreground intensity bit (3) can be programmed to switch between character sets A and B on EGA and VGA cards, thus enabling 512 simultaneous characters on screen. In this case the bit's usual function (intensity) is regularly turned off.

in graphics modes, only characters drawn with white foreground pixels are matched by the pattern-comparison routine

on the Tandy 2000, BH=FFh specifies that the current page should be

used

because of the IBM BIOS specifications, there may exist some clone BIOSes which do not preserve SI or DI; the Novell DOS kernel preserves SI, DI, and BP before many INT 10h calls to avoid problems due to those registers not being preserved by the BIOS.

BUG: some IBM PC ROM BIOSes destroy BP when in graphics modes

SeeAlso: AH=09h,AX=1003h,AX=1103h,AH=12h/BL=37h,AX=5001h

Bitfields for character's display attribute:

Bit(s) Description (Table 00014)

7 foreground blink or (alternate) background bright (see also AX=1003h)

6-4 background color (see #00015)

3 foreground bright or (alternate) alternate character set (see AX=1103h)

2-0 foreground color (see #00015)

SeeAlso: #00026

(Table 00015)

Values for character color:

Normal	Bright
000b black	dark gray
001b blue	light blue
010b green	light green
011b cyan	light cyan
100b red	light red
101b magenta	light magenta
110b brown	yellow
111b light gray	white

-----V-1009-----

INT 10 - VIDEO - WRITE CHARACTER AND ATTRIBUTE AT CURSOR POSITION

AH = 09h

AL = character to display

BH = page number (00h to number of pages - 1) (see #00010)

background color in 256-color graphics modes (ET4000)

BL = attribute (text mode) or color (graphics mode)

if bit 7 set in <256-color graphics mode, character is XOR'ed onto screen

CX = number of times to write character

Return: nothing

Notes: all characters are displayed, including CR, LF, and BS

replication count in CX may produce an unpredictable result in graphics

modes if it is greater than the number of positions remaining in the

current row

With PhysTechSoft's PTS ROM-DOS the BH, BL, and CX values are ignored on entry.

SeeAlso: AH=08h,AH=0Ah,AH=4Bh"GRAFIX",INT 17/AH=60h,INT 1F"SYSTEM DATA"

SeeAlso: INT 43"VIDEO DATA",INT 44"VIDEO DATA"

-----V-100A-----

INT 10 - VIDEO - WRITE CHARACTER ONLY AT CURSOR POSITION

AH = 0Ah

AL = character to display

BH = page number (00h to number of pages - 1) (see #00010)

background color in 256-color graphics modes (ET4000)

BL = attribute (PCjr, Tandy 1000 only) or color (graphics mode)

if bit 7 set in <256-color graphics mode, character is XOR'ed onto screen

CX = number of times to write character

Return: nothing

Notes: all characters are displayed, including CR, LF, and BS replication count in CX may produce an unpredictable result in graphics modes if it is greater than the number of positions remaining in the current row

With PhysTechSoft's PTS ROM-DOS the BH and CX values are ignored on entry.

SeeAlso: AH=08h,AH=09h,AH=11h"Tandy 2000",AH=4Bh,INT 17/AH=60h

SeeAlso: INT 1F"SYSTEM DATA",INT 43"VIDEO DATA",INT 44"VIDEO DATA"

-----V-100B--BH00-----

INT 10 - VIDEO - SET BACKGROUND/BORDER COLOR

AH = 0Bh

BH = 00h

BL = background/border color (border only in text modes)

Return: nothing

SeeAlso: AH=0Bh/BH=01h

-----V-100B--BH01-----

INT 10 - VIDEO - SET PALETTE

AH = 0Bh

BH = 01h

BL = palette ID

00h background, green, red, and brown/yellow

01h background, cyan, magenta, and white

Return: nothing

Note: this call was only valid in 320x200 graphics on the CGA, but newer cards support it in many or all graphics modes

SeeAlso: AH=0Bh/BH=00h, MEM 0040h:0066h

-----V-100B--BH02-----

INT 10 - Tandy 2000 - VIDEO - SET PALETTE ENTRY

AH = 0Bh

BH = 02h

BL = palette entry number

AL = new color value for palette entry

Return: nothing

Note: this interrupt is identical to INT 52 for Tandy 2000

SeeAlso: INT 10/AH=00h, INT 10/AH=0Bh/BH=01h, INT 52"Tandy 2000"

-----V-100C-----

INT 10 - VIDEO - WRITE GRAPHICS PIXEL

AH = 0Ch

BH = page number

AL = pixel color

if bit 7 set, value is XOR'ed onto screen except in 256-color modes

CX = column

DX = row

Return: nothing

Desc: set a single pixel on the display in graphics modes

Notes: valid only in graphics modes

BH is ignored if the current video mode supports only one page

SeeAlso: AH=0Dh, AH=46h

-----V-100D-----

INT 10 - VIDEO - READ GRAPHICS PIXEL

AH = 0Dh

BH = page number

CX = column

DX = row

Return: AL = pixel color

Desc: determine the current color of the specified pixel in graphics modes

Notes: valid only in graphics modes

BH is ignored if the current video mode supports only one page

SeeAlso: AH=0Ch, AH=47h

-----V-100E-----

INT 10 - VIDEO - TELETYPE OUTPUT

AH = 0Eh

AL = character to write

BH = page number

BL = foreground color (graphics modes only)

Return: nothing

Desc: display a character on the screen, advancing the cursor and scrolling
the screen as necessary

Notes: characters 07h (BEL), 08h (BS), 0Ah (LF), and 0Dh (CR) are interpreted
and do the expected things

IBM PC ROMs dated 1981/4/24 and 1981/10/19 require that BH be the same
as the current active page

BUG: if the write causes the screen to scroll, BP is destroyed by BIOSes
for which AH=06h destroys BP

SeeAlso: AH=02h,AH=06h,AH=0Ah

-----b-100E--CXABCD-----

INT 10 - V20-XT-BIOS - TELETYPE OUTPUT WITH ATTRIBUTE

AH = 0Eh
CX = ABCDh
BP = ABCDh
AL = character to write
BH = page number
BL = foreground color (text modes as well as graphics modes)

Return: nothing

Program: V20-XT-BIOS is a ROM BIOS replacement with extensions by Peter
Koehlmann / c't magazine

Desc: display a character on the screen, advancing the cursor and scrolling
the screen as necessary

Notes: characters 07h (BEL), 08h (BS), 0Ah (LF), and 0Dh (CR) are interpreted
and do the expected things

SeeAlso: INT 15/AH=84h"V20-XT-BIOS"

-----V-100F-----

INT 10 - VIDEO - GET CURRENT VIDEO MODE

AH = 0Fh
Return: AH = number of character columns
AL = display mode (see #00010 at AH=00h)
BH = active page (see AH=05h)

Notes: if mode was set with bit 7 set ("no blanking"), the returned mode will
also have bit 7 set

EGA, VGA, and UltraVision return either AL=03h (color) or AL=07h
(monochrome) in all extended-row text modes

HP 200LX returns AL=07h (monochrome) if mode was set to AL=21h
and always 80 resp. 40 columns in all text modes regardless of
current zoom setting (see AH=D0h)

when using a Hercules Graphics Card, additional checks are necessary:
mode 05h: if WORD 0040h:0063h is 03B4h, may be in graphics page 1
(as set by DOSSHELL and other Microsoft software)

mode 06h: if WORD 0040h:0063h is 03B4h, may be in graphics page 0

(as set by DOSSHELL and other Microsoft software)

mode 07h: if BYTE 0040h:0065h bit 1 is set, Hercules card is in graphics mode, with bit 7 indicating the page (mode set by Hercules driver for Borland Turbo C)

the Tandy 2000 BIOS is only documented as returning AL, not AH or BH

SeeAlso: AH=00h,AH=05h,AX=10F2h,AX=1130h,AX=CD04h,MEM 0040h:004Ah

-----V-100F56BX4756-----

INT 10 - VUIMAGE DISPLAY DRIVER (v2.20 and below)

AX = 0F56h

BX = 4756h

CX = 4944h

DL = function

01h installation check

Return: AX = 5649h

BX = 4443h

CX = 5647h

DH = 01h

02h get first video mode's parameters

Return: AX = BIOS mode number

BX = width in pixels

CX = height in pixels

DX = number of colors

03h get next video mode's parameters

Return: as for DL=02h

04h display line???

ES:DI -> record (see #00016)

???

Return: ???

Program: VUIMAGE is a shareware GIF/TIFF image viewer by Offe Enterprises

Note: the use of TSR display drivers was discontinued after v2.20

Index: installation check;VUIMAGE display driver

Format of record for VUIMAGE Function 04h:

Offset Size Description (Table 00016)

00h WORD row number

02h WORD starting column???

04h WORD ending column???

???

-----V-100F--SIF123-----

INT 10 - FRIEZE v7.41+ - INSTALLATION CHECK

AH = 0Fh
SI = F123h
DI = 321Fh

Return: AH = number of character columns

AL = display mode (see #00010 at AH=00h)
BH = active page (see AH=05h)
SI = DI = F345h if installed

Notes: if mode was set with bit 7 set ("no blanking"), the returned mode will also have bit 7 set

EGA, VGA, and UltraVision return either AL=03h (color) or AL=07h (monochrome) in all extended-row text modes

SeeAlso: AH=0Fh"VIDEO",AH=4Bh"FRIEZE"

-----V-1010-----

INT 10 - BIOS Window Extension v1.1 - SET WINDOW COORDINATES

AH = 10h
CH,CL = row,column of upper left corner of window
DH,DL = row,column of lower right corner of window

Return: AL = status

00h successful
01h failed

AH destroyed

Program: BWE is a TSR by John J. Seal published in May 1986 Dr. Dobb's Journal

Note: when a window has been set, all output via AH=0Eh is restricted to the specified window

SeeAlso: AH=11h"Window",AH=12h"Window"

-----V-1010-----

INT 10 - VIDEO - Eagle PC2 BIOS Rev. C - SET SCROLL SPEED

AH = 10h
AL = speed
00h fast
01h slow (scrolling only moves characters during vertical retrace)

Return: AH = previous speed

-----V-1010-----

INT 10 - Tandy 2000 - VIDEO - GET/SET CHARACTER FONTS

AH = 10h
AL = control value
bit 0: set character set instead of reading it
bit 1: high 128 characters instead of low 128 characters

ES:BX -> new character set if AL bit 0 set

Return: ES:BX -> current character set if AL bit 0 clear on entry

Notes: this interrupt is identical to INT 52 on Tandy 2000

the character set consists of 16 bytes for each of the 128 characters,
where each of the 16 bytes describes the pixels in one scan line,
most significant bit leftmost

SeeAlso: AH=00h,AH=0Bh/BH=02h,AH=11h"Tandy 2000",AH=12h"Tandy 2000"

SeeAlso: INT 52"Tandy 2000"

-----V-101000-----

INT 10 - VIDEO - SET SINGLE PALETTE REGISTER (PCjr,Tandy,EGA,MCGA,VGA)

AX = 1000h

BL = palette register number (00h-0Fh)

= attribute register number (undocumented) (see #00017)

BH = color or attribute register value

Return: nothing

Notes: on MCGA, only BX = 0712h is supported

under UltraVision, the palette locking status (see AX=CD01h)

determines the outcome

SeeAlso: AX=1002h,AX=1007h,AX=CD01h

(Table 00017)

Values for attribute register number:

10h attribute mode control register (should let BIOS control this)

11h overscan color register (see also AX=1001h)

12h color plane enable register (bits 3-0 enable corresponding
text attribute bit)

13h horizontal PEL panning register

14h color select register

-----V-101001-----

INT 10 - VIDEO - SET BORDER (OVERSCAN) COLOR (PCjr,Tandy,EGA,VGA)

AX = 1001h

BH = border color (00h-3Fh)

Return: nothing

BUG: the original IBM VGA BIOS incorrectly updates the parameter save area

and places the border color at offset 11h of the palette table

rather than offset 10h

Note: under UltraVision, the palette locking status (see AX=CD01h)

determines the outcome

SeeAlso: AX=1002h,AX=1008h,AX=CD01h

-----V-101002-----

INT 10 - VIDEO - SET ALL PALETTE REGISTERS (PCjr,Tandy,EGA,VGA)

AX = 1002h

ES:DX -> palette register list (see #00018)

BH = 00h to avoid problems on some adapters

Return: nothing

Note: under UltraVision, the palette locking status (see AX=CD01h) determines the outcome

SeeAlso: AX=1000h,AX=1001h,AX=1009h,AX=CD01h

Format of palette register list:

Offset Size Description (Table 00018)

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

10h BYTE border color

SeeAlso: #00461

-----V-101003-----

INT 10 - VIDEO - TOGGLE INTENSITY/BLINKING BIT (Jr, PS, TANDY 1000, EGA, VGA)

AX = 1003h

BL = new state

00h background intensity enabled

01h blink enabled

BH = 00h to avoid problems on some adapters

Return: nothing

Notes: although there is no function to get the current status on adapters prior to the VGA, bit 5 of 0040h:0065h indicates the state; on the VGA, use AH=1Bh and check offset 2Dh of the returned data (see #00040) when configured for a monochrome display, the Boca Research Multi-EGA with ROM v M1.1 Type D has its screen disrupted if BH is not clear this call is reported to be "Get Cursor Position" on the Tandy 1000SL/TL

SeeAlso: AH=08h, AH=1Bh

-----V-101007-----

INT 10 - VIDEO - GET INDIVIDUAL PALETTE REGISTER (VGA,UltraVision v2+)

AX = 1007h

BL = palette or attribute (undoc) register number (see #00017)

Return: BH = palette or attribute register value

Note: UltraVision v2+ supports this function even on color EGA systems in video modes 00h-03h, 10h, and 12h; direct programming of the palette registers will cause incorrect results because the EGA registers are write-only. To guard against older versions or unsupported video modes, programs which expect to use this function on EGA systems should set BH to FFh on entry.

SeeAlso: AX=1000h,AX=1009h

-----V-101008-----

INT 10 - VIDEO - READ OVERSCAN (BORDER COLOR) REGISTER (VGA,UltraVision v2+)

AX = 1008h

Return: BH = border color (00h-3Fh)

Note: (see AX=1007h)

SeeAlso: AX=1001h

-----V-101009-----

INT 10 - VIDEO - READ ALL PALETTE REGISTERS AND OVERSCAN REGISTER (VGA)

AX = 1009h

ES:DX -> 17-byte buffer for palette register list (see #00018)

Return: nothing

Note: UltraVision v2+ supports this function even on color EGA systems in video modes 00h-03h, 10h, and 12h; direct programming of the palette registers will cause incorrect results because the EGA registers are write-only. To guard against older versions or unsupported video modes, programs which expect to use this function on EGA systems should set the ES:DX buffer to FFh before calling.

SeeAlso: AX=1002h,AX=1007h,AX=CD02h

-----V-101010-----

INT 10 - VIDEO - SET INDIVIDUAL DAC REGISTER (VGA/MCGA)

AX = 1010h

BX = register number

CH = new value for green (0-63)

CL = new value for blue (0-63)

DH = new value for red (0-63)

Return: nothing

SeeAlso: AX=1012h,AX=1015h

-----V-101012-----

INT 10 - VIDEO - SET BLOCK OF DAC REGISTERS (VGA/MCGA)

AX = 1012h

BX = starting color register

CX = number of registers to set

ES:DX -> table of 3*CX bytes where each 3 byte group represents one byte each of red, green and blue (0-63)

Return: nothing

SeeAlso: AX=1010h,AX=1017h,INT 62/AX=00A5h

-----V-101013-----

INT 10 - VIDEO - SELECT VIDEO DAC COLOR PAGE (VGA)

AX = 1013h

BL = subfunction

00h select paging mode

BH = 00h select 4 blocks of 64

BH = 01h select 16 blocks of 16

01h select page

BH = page number (00h to 03h) or (00h to 0Fh)

Return: nothing

Note: this function is not valid in mode 13h

SeeAlso: AX=101Ah

-----V-101015-----

INT 10 - VIDEO - READ INDIVIDUAL DAC REGISTER (VGA/MCGA)

AX = 1015h

BL = palette register number

Return: DH = red value

CH = green value

CL = blue value

AX destroyed by some BIOSes

(e.g. Tseng ET4000 BIOS v8.00n always returns AX=00C9h)

SeeAlso: AX=1010h,AX=1017h

-----V-101017-----

INT 10 - VIDEO - READ BLOCK OF DAC REGISTERS (VGA/MCGA)

AX = 1017h

BX = starting palette register

CX = number of palette registers to read

ES:DX -> buffer (3 * CX bytes in size) (see also AX=1012h)

Return: buffer filled with CX red, green and blue triples

SeeAlso: AX=1012h,AX=1015h,INT 62/AX=00A4h

-----V-101018-----

INT 10 U - VIDEO - SET PEL MASK (VGA/MCGA)

AX = 1018h

BL = new PEL value

Return: nothing

SeeAlso: AX=1019h

-----V-101019-----

INT 10 U - VIDEO - READ PEL MASK (VGA/MCGA)

AX = 1019h

Return: BL = value read

SeeAlso: AX=1018h

-----V-10101A-----

INT 10 - VIDEO - GET VIDEO DAC COLOR-PAGE STATE (VGA)

AX = 101Ah

Return: BL = paging mode

00h four pages of 64

01h sixteen pages of 16

BH = current page

SeeAlso: AX=1013h

-----V-10101B-----

INT 10 - VIDEO - PERFORM GRAY-SCALE SUMMING (VGA/MCGA)

AX = 101Bh

BX = starting palette register

CX = number of registers to convert

Return: nothing

Desc: convert the RGB values of one or more palette registers such that the resulting values are grays with the same intensities as the original colors

SeeAlso: AH=12h/BL=33h

-----V-1010E0-----

INT 10 - VIDEO - Diamond Speedstar 24 - SET 24-BIT GRAPHICS MODE

AX = 10E0h

BL = video mode (see also #00010 at AH=00h)

2Eh = 640x480

Return: ???

SeeAlso: AH=00h,AX=10F0h

-----V-1010F0-----

INT 10 - VIDEO - Tseng ET-4000 BIOS - SET HiColor GRAPHICS MODE

AX = 10F0h

BL = video mode (see also #00010 at AH=00h)

13h = 320x200x32K

2Dh = 640x350x32K

2Eh = 640x480x32K

2Fh = 640x400x32K

30h = 800x600x32K

3Eh = 640x480x16M (Genoa 7900)

FFh Tseng 24-bit color mode

BH = mode

2Dh = 640x480

2Eh = 640x480x16M

2Fh = 640x400

30h = 800x600

38h = 1024x768??? (Tseng ET4000/W32i)

Return: AL = 10h if supported

AH = status

00h if successful

other on error

Desc: switch the display into a graphics mode with 15 or more bits per pixel

Note: the Tseng HiColor BIOS extensions are supported by:

Diamond Computer Systems SpeedStar HiColor VGA

Everex Systems HC VGA
Focus Information Systems 2theMax 4000
Cardinal Technologies VGA732
Orchid ProDesigner IIs Genoa 7900

SeeAlso: AH=00h,AX=10E0h,AX=10F1h,AX=10F2h

Index: video modes;Tseng Hi-Color

-----V-1010F1-----

INT 10 - VIDEO - Tseng ET-4000 BIOS - GET DAC TYPE

AX = 10F1h

Return: AL = 10h if supported

BL = type of digital/analog converter (see #00019)

Desc: determine which type of digital-to-analog converter is installed on the
display board

SeeAlso: AX=10F0h,AX=10F2h

(Table 00019)

Values for type of DAC:

00h normal VGA DAC

01h Sierra SC1148x HiColor DAC

---Diamond SpeedStar 24---

02h new Sierra SS24 DAC (24-bit)

---generic Tseng BIOS v8+ ---

02h Sierra Mark2 (15-bit) or Mark3 (15/16-bit) DAC

03h ATT20c490/1/2 (15/16/24-bit)

04h AcuMos ADAC1 (15/16/24-bit)

05h unknown 15/16/24-bit DAC

06h Cirrus Internal 15/16/24-bit (CL-GD54xx series adapters)

07h Diamond SS2410 (15/24-bit)

08h unknown 15/16/24-bit DAC

09h unknown 15/16/24-bit DAC

else other HiColor DAC

-----V-1010F2-----

INT 10 u - VIDEO - Tseng ET-4000 BIOS - CHECK IF HiColor MODE/SET HiColor MODE

AX = 10F2h

BL = function

00h get current HiColor mode

01h set 15-bit HiColor mode

02h set 16-bit HiColor mode

Return: AX = 0010h if supported

BL = video mode type

00h not in HiColor mode

01h 15-bit RGB mode

02h 16-bit RGB mode

03h 24-bit RGB mode

Desc: determine whether the display is in a graphics mode with 15 or more bits per pixel color resolution

Note: set (BL=01h/02h) only works if already in a HiColor mode

SeeAlso: AH=0Fh,AX=10F0h,AX=10F1h

-----V-1011-----

INT 10 - BIOS Window Extension v1.1 - GET WINDOW COORDINATES

AH = 11h

Return: CH,CL = row,column of upper left corner

DH,DL = row,column of lower right corner

Desc: determine the portion of the display to which output is restricted

Program: BWE is a TSR by John J. Seal published in May 1986 Dr. Dobb's Journal

SeeAlso: AH=10h"Window",AH=12h"Window"

-----V-1011-----

INT 10 - Tandy 2000 - VIDEO - WRITE ATTRIBUTE ONLY

AH = 11h

BL = new character attribute (text modes) or color (graphics)

CX = number of times to write attribute

Return: nothing

Note: this interrupt is identical to INT 52 on Tandy 2000

SeeAlso: AH=0Ah,AH=0Bh/BH=02h,AH=11h"Tandy 2000",AH=12h"Tandy 2000"

SeeAlso: INT 52"Tandy 2000"

-----V-101100-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD USER-SPECIFIED PATTERNS (PS, EGA, VGA)

AX = 1100h

ES:BP -> user table

CX = count of patterns to store

DX = character offset into map 2 block

BL = block to load in map 2

BH = number of bytes per character pattern

Return: nothing

Notes: This function will cause a mode set, completely resetting the video environment, but without clearing the video buffer the current block specifiers may be determined with INT 10/AH=1Bh, looking at offsets 2Bh and 2Ch of the returned data (VGA only) (see AH=1Bh,#00040)

SeeAlso: AX=1101h,AX=1102h,AX=1103h,AX=1104h,AX=1110h,AH=1Bh,AX=CD10h

SeeAlso: MEM 0040h:0084h

Index: text mode;font|text mode;screen rows

-----V-101101-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD ROM MONOCHROME PATTERNS (PS, EGA, VGA)

AX = 1101h

BL = block to load

Return: nothing

Notes: (see also AX=1100h)

This function will cause a mode set, completely resetting
the video environment, but without clearing the video buffer
the "monochrome" patterns are 8x14 pixels in size

SeeAlso: AX=1100h, AX=1102h, AX=1103h, AX=1104h, AX=1111h, AH=1Bh, AX=CD10h

SeeAlso: MEM 0040h:0084h

Index: text mode;font|text mode;screen rows

-----V-101102-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD ROM 8x8 DBL-DOT PATTERNS (PS, EGA, VGA)

AX = 1102h

BL = block to load

Return: nothing

Notes: (see AX=1100h)

SeeAlso: AX=1100h, AX=1101h, AX=1103h, AX=1104h, AX=1112h, AH=1Bh, AX=CD10h

SeeAlso: MEM 0040h:0084h

-----V-101103-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - SET BLOCK SPECIFIER (PS, EGA, VGA)

AX = 1103h

BL = block specifier (see #00020)

Return: nothing

Notes: (see also AX=1110h)

This function allows dual character sets to appear on screen
simultaneously

SeeAlso: AX=1100h, AX=1101h, AX=1102h, AX=1104h, AH=1Bh, AX=CD10h

SeeAlso: MEM 0040h:0084h

Index: text mode;font|text mode;screen rows

Bitfields for block specifier:

Bit(s) Description (Table 00020)

---EGA/MCGA---

0,1 block selected by characters with attribute bit 3 clear

2,3 block selected by characters with attribute bit 3 set

---VGA---

0,1,4 block selected by characters with attribute bit 3 clear

2,3,5 block selected by characters with attribute bit 3 set

-----V-101104-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD ROM 8x16 CHARACTER SET (VGA)

AX = 1104h

BL = block to load

Return: nothing

Notes: (see AX=1100h)

SeeAlso: AX=1100h,AX=1101h,AX=1102h,AX=1103h,AX=1114h,AH=1Bh,AX=CD10h

SeeAlso: MEM 0040h:0084h

Index: text mode;font|text mode;screen rows

-----V-1011-----

INT 10 - VIDEO - Realtek RTVGA - TEXT-MODE CHARACTER GENERATOR FUNCTIONS

AH = 11h

AL = 07h load user-specified patterns and recalculate mode parms

parameters are the same as for AX=1110h

AL = 08h load monochrome patterns (8x14) and recalculate mode parms

parameters are the same as for AX=1111h

AL = 09h load ROM 8 by 8 double-dot patterns and recalculate mode parms

parameters are the same as for AX=1112h

AL = 0Bh load ROM 8x16 character set (VGA) and recalculate mode parms

parameters are the same as for AX=1114h

Note: these functions should only be called under the same conditions as

AL=1xh (see AX=1110h"EGA")

SeeAlso: AX=1100h,AX=1110h"CHARGEN",AX=1110h"Realtek",MEM 0040h:0084h

Index: text mode;font|text mode;screen rows

-----V-10110F-----

INT 10 - VIDEO - Realtek RTVGA - SET USER 8x8 GRAPHICS CHARACTERS

AX = 110Fh

ES:BP -> user table for INT 1F

Return: nothing

Note: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

SeeAlso: AX=1110h"Realtek",AX=1111h"Realtek",AX=1112h"Realtek",AX=1120h

-----V-101110-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD USER-SPECIFIED PATTERNS (PS, EGA, VGA)

AX = 1110h

ES:BP -> user table

CX = count of patterns to store

DX = character offset into map 2 block

BL = block to load in map 2

BH = number of bytes per character pattern

Return: nothing

Notes: This function will cause a mode set, completely resetting

the video environment, but without clearing the video buffer

This function is designed to be called immediately after a mode set,
it is equivalent to AX=110xh except that:

Page 0 must be active.

Bytes/character is recalculated.

Max character rows is recalculated.

CRT buffer length is recalculated.

CRTC registers are reprogrammed as follows:

R09 = bytes/char-1 ; max scan line (mode 7 only)

R0A = bytes/char-2 ; cursor start

R0B = 0 ; cursor end

R12 = ((rows+1)*(bytes/char))-1 ; vertical display end

R14 = bytes/char ; underline loc

(*** BUG: should be 1 less ***)

the current block specifiers may be determined with INT 10/AH=1Bh,
looking at offsets 2Bh and 2Ch of the returned data (VGA only)
(see AH=1Bh,#00040)

SeeAlso: AX=1100h,AX=1111h,AX=1112h,AX=1114h,AH=1Bh,AX=CD10h,MEM 0040h:0084h

Index: text mode;font|text mode;screen rows

-----V-101110-----

INT 10 - VIDEO - Realtek RTVGA - SET USER GRAPHICS CHARACTERS

AX = 1110h

ES:BP -> user table

CX = bytes per character

BL = row specifier

00h user set

DL = number of rows

01h 14 rows

02h 25 rows

03h 43 rows

Return: nothing

Note: this function is meant to be called immediately after a mode set;
results are unpredictable at other times

SeeAlso: AX=110Fh"Realtek",AX=1112h"Realtek",AX=1121h

-----V-101111-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD ROM MONOCHROME PATTERNS (PS,EGA,VGA)

AX = 1111h

BL = block to load

Return: nothing

Notes: (see AX=1110h)

the "monochrome" patterns are 8x14 pixels in size

SeeAlso: AX=1101h,AX=1110h,AX=1112h,AX=1114h,AH=1Bh,AX=CD10h

-----V-101111-----

INT 10 - VIDEO - Realtek RTVGA - SET ROM 8x14 GRAPHICS CHARACTERS

AX = 1111h

BL = row specifier (see AX=1121h)

Return: nothing

Note: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

SeeAlso: AX=110Fh"Realtek",AX=1110h"Realtek",AX=1122h

-----V-101112-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD ROM 8x8 DBL-DOT PATTERNS (PS,EGA,VGA)

AX = 1112h

BL = block to load

Return: nothing

Notes: (see AX=1110h)

SeeAlso: AX=1103h,AX=1110h,AX=1111h,AX=1114h,AH=1Bh,AX=CD10h

-----V-101112-----

INT 10 - VIDEO - Realtek RTVGA - SET ROM 8x8 DOUBLE-DOT CHARACTERS

AX = 1112h

BL = row specifier (see AX=1121h)

Return: nothing

Note: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

SeeAlso: AX=110Fh"Realtek",AX=1110h"Realtek",AX=1111h"Realtek",AX=1123h

-----V-101114-----

INT 10 - VIDEO - TEXT-MODE CHARGEN - LOAD ROM 8x16 CHARACTER SET (VGA)

AX = 1114h

BL = block to load

Return: nothing

Notes: (see AX=1110h)

SeeAlso: AX=1104h,AX=1110h,AX=1111h,AX=1112h,AH=1Bh,AX=CD10h

-----V-101118-----

INT 10 - IBM SurePath BIOS - Officially "Private" Function

AX = 1118h

-----V-101120-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - SET USER 8x8 GRAPHICS CHARS (PS,EGA,VGA)

AX = 1120h

ES:BP -> user table for INT 1F

Return: nothing

Note: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

SeeAlso: AX=1121h,AX=1122h,AX=1123h,AX=1124h,AX=1129h,INT 1F"SYSTEM DATA"

SeeAlso: INT 43"VIDEO DATA"

-----V-101121-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - SET USER GRAPHICS CHARACTERS (PS,EGA,VGA)

AX = 1121h

ES:BP -> user table

CX = bytes per character

BL = row specifier

00h user set

DL = number of rows

01h 14 rows

02h 25 rows

03h 43 rows

Return: AL = new number of rows (Diamond Stealth64 Video)

Note: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

SeeAlso: AX=1120h,AX=1122h,AX=1123h,AX=1124h,AX=1129h"Compaq"

SeeAlso: AX=1129h"Diamond",INT 1F"SYSTEM DATA",INT 43"VIDEO DATA"

-----V-101122-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - SET ROM 8x14 GRAPHICS CHARS (PS,EGA,VGA)

AX = 1122h

BL = row specifier (see AX=1121h)

Return: nothing

Notes: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

UltraVision v2+ sets INT 43 to the appropriate font on this call

SeeAlso: AX=1111h,AX=1120h,AX=1121h,AX=1123h,AX=1124h,AX=1129h

SeeAlso: INT 1F"SYSTEM DATA",INT 43"VIDEO DATA"

-----V-101123-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - SET ROM 8x8 DOUBLE-DOT CHARS (PS,EGA,VGA)

AX = 1123h

BL = row specifier (see AX=1121h)

Return: nothing

Notes: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

UltraVision v2+ sets INT 43 to the appropriate font on this call

SeeAlso: AX=1112h,AX=1120h,AX=1121h,AX=1122h,AX=1124h,AX=1129h

SeeAlso: INT 1F"SYSTEM DATA",INT 43"VIDEO DATA"

-----V-101124-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - LOAD 8x16 GRAPHICS CHARS (VGA,MCGA)

AX = 1124h

BL = row specifier (see AX=1121h)

Return: nothing

Notes: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

UltraVision v2+ sets INT 43 to the appropriate font on this call

SeeAlso: AX=1120h,AX=1121h,AX=1122h,AX=1123h,AX=1129h

SeeAlso: INT 1F"SYSTEM DATA",INT 43"VIDEO DATA"

-----V-101129-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - LOAD 8x16 GRAPH CHARS (Compaq Systempro)

AX = 1129h

BL = row specifier (see AX=1121h)

Return: nothing

Notes: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

UltraVision v2+ sets INT 43 to the appropriate font on this call

SeeAlso: AX=1120h,AX=1121h,AX=1122h,AX=1123h,AX=1124h,INT 1F"SYSTEM DATA"

SeeAlso: INT 43"VIDEO DATA"

-----V-101129-----

INT 10 - VIDEO - GRAPH-MODE CHARGEN - SET USER GRAPHICS CHARACTERS (Diamond)

AX = 1129h

DI:BP -> user table

CX = bytes per character

BL = row specifier

00h user set

DL = number of rows

01h 14 rows

02h 25 rows

03h 43 rows

Return: AL = new number of rows

Notes: this function is meant to be called immediately after a mode set;

results are unpredictable at other times

supported by Diamond Stealth64 Video

SeeAlso: AX=1120h,AX=1122h,AX=1123h,AX=1124h,INT 1F"SYSTEM",INT 43"VIDEO"

-----V-101130-----

INT 10 - VIDEO - GET FONT INFORMATION (EGA, MCGA, VGA)

AX = 1130h

BH = pointer specifier

00h INT 1Fh pointer

01h INT 43h pointer

02h ROM 8x14 character font pointer

03h ROM 8x8 double dot font pointer

04h ROM 8x8 double dot font (high 128 characters)
 05h ROM alpha alternate (9 by 14) pointer (EGA,VGA)
 06h ROM 8x16 font (MCGA, VGA)
 07h ROM alternate 9x16 font (VGA only) (see #00021)
 11h (UltraVision v2+) 8x20 font (VGA) or 8x19 font (autosync EGA)
 12h (UltraVision v2+) 8x10 font (VGA) or 8x11 font (autosync EGA)

Return: ES:BP = specified pointer

CX = bytes/character of on-screen font (not the requested font!)
 DL = highest character row on screen

Note: for UltraVision v2+, the 9xN alternate fonts follow the corresponding
 8xN font at ES:BP+256N

BUG: the IBM EGA and some other EGA cards return in DL the number of rows on
 screen rather than the highest row number (which is one less).

SeeAlso: AX=1100h,AX=1103h,AX=1120h,INT 1F"SYSTEM DATA",INT 43"VIDEO DATA"

Format of alternate font table [array]:

Offset Size Description (Table 00021)

00h BYTE character to be replaced (00h = end of table)
 01h N BYTES graphics data for character, one byte per scan line

-----V-101130BX4D4F-----

INT 10 - M10_SCR.COM v3.5+ - INSTALLATION CHECK

AX = 1130h
 BX = 4D4Fh

Return: CX = 4F4Dh if installed

ES:BP -> M10_SCR INT 10 handler

Program: M10_SCR is a shareware extended text mode and font driver by
 I.V. Morozov

SeeAlso: INT 16/AX=4D4Fh

-----V-1012-----

INT 10 - BIOS Window Extension v1.1 - GET BLANKING ATTRIBUTE

AH = 12h

Return: BH = attribute to use on blanked lines when scrolling

Program: BWE is a TSR by John J. Seal published in May 1986 Dr. Dobb's Journal

SeeAlso: AH=11h"Window",AH=12h"Window"

-----V-1012-----

INT 10 - Tandy 2000 - VIDEO - SCROLL WINDOW

AH = 12h

AL = number of rows or columns to scroll

BH = buffer flag

00h data in user buffer

ES:SI -> buffer containing character/attribute pairs

01h no buffer, fill emptied rows/columns with blanks

BL = direction in which to scroll

00h up

01h down

02h left

03h right

CH,CL = row,column of upper left corner of scroll area

DH,DL = row,column of lower right corner

Return: nothing

Notes: this interrupt is identical to INT 52 on Tandy 2000

the user buffer, if supplied, must be organized by row; regardless of the scroll direction, all character/attribute pairs for the first row are first, then the pairs for the second row, etc.

SeeAlso: AH=00h,AH=0Bh/BH=02h,AH=11h"Tandy 2000",AH=12h"Tandy 2000"

SeeAlso: INT 52"Tandy"

-----V-1012--BL10-----

INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (PS, EGA, VGA, MCGA) - GET EGA INFO

AH = 12h

BL = 10h

Return: BH = video state

00h color mode in effect (I/O port 3Dxh)

01h mono mode in effect (I/O port 3Bxh)

BL = installed memory (00h = 64K, 01h = 128K, 02h = 192K, 03h = 256K)

CH = feature connector bits (see #00022)

CL = switch settings (see #00023,#00024)

AH destroyed (at least by Tseng ET4000 BIOS v8.00n)

Note: one possible check for the presence of an EGA or later display card is to call this function with BH=FFh; if not present, BH will be unchanged on return

another installation check is used by Athena Digital's HGCIBM, which sets CX to FFFFh on calling and checks whether it has been changed on return

Index: installation check;EGA

Bitfields for feature connector bits:

Bit(s) Description (Table 00022)

0 FEAT 1 line, state 2

1 FEAT 0 line, state 2

2 FEAT 1 line, state 1

3 FEAT 0 line, state 1

4-7 unused (0)

Bitfields for switch settings:

Bit(s) Description (Table 00023)

0 switch 1 OFF
1 switch 2 OFF
2 switch 3 OFF
3 switch 4 OFF
4-7 unused

(Table 00024)

Values for switch settings on original EGA/VGA:

00h primary MDA/HGC, secondary EGA+ 40x25
01h-03h primary MDA/HGC, secondary EGA+ 80x25
04h primary CGA 40x25, secondary EGA+ 80x25 mono
05h primary CGA 80x25, secondary EGA+ 80x25 mono
06h primary EGA+ 40x25, secondary MDA/HGC (optional)
07h-09h primary EGA+ 80x25, secondary MDA/HGC (optional)
0Ah primary EGA+ 80x25 mono, secondary CGA 40x25 (optional)
0Bh primary EGA+ 80x25 mono, secondary CGA 80x25 (optional)

-----V-1012--BL11-----

INT 10 - VIDEO - Trident TVGA 8800/8900/9000 BIOS - GET BIOS INFO

AH = 12h

BL = 11h

Return: AL = 12h if function supported

BL = ??? (10h)

ES:BP -> BIOS info structure (see #00025)

SeeAlso: AH=12h/BL=12h"TRIDENT"

Format of Trident BIOS Info structure:

Offset Size Description (Table 00025)

00h BYTE ??? (0000h)
01h BYTE OEM Code (00h for original Trident)
02h WORD ID??? (1073h for 8800BR, 1074 for 8800CS)
04h 8 BYTES BIOS date ('mm/dd/yy')
0Ch WORD ???
0Eh 8 BYTES BIOS Version (' C3-128 ', ' C3-129 ')

-----V-1012--BL12-----

INT 10 - VIDEO - Trident TVGA 8800/8900/9000 BIOS - GET VIDEO RAM SIZE

AH = 12h

BL = 12h

Return: AL = 12h if function supported

AH = number of 256K banks of RAM installed
(read from CRTC register 1Fh)

SeeAlso: AH=12h/BL=11h"TRIDENT"

-----V-1012--BL20-----

INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (PS,EGA,VGA,MCGA) - ALTERNATE PRtSC

AH = 12h

BL = 20h select alternate print screen routine

Return: nothing

Desc: installs a PrtSc routine from the video card's BIOS to replace the
default PrtSc handler from the ROM BIOS, which usually does not
understand screen heights other than 25 lines

Note: some adapters disable print-screen instead of enhancing it

SeeAlso: INT 05"PRINT SCREEN"

-----V-1012--BL2E-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG

AH = 12h

BL = 2Eh

Return: nothing

Note: due to an omitted end-of-list marker, these versions of the BIOS will
crash the system on this function

-----V-1012--BL30-----

INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (VGA) - SELECT VERTICAL RESOLUTION

AH = 12h

BL = 30h

AL = vertical resolution

00h 200 scan lines

01h 350 scan lines

02h 400 scan lines

Return: AL = 12h if function supported

Desc: specify the number of scan lines used to display text modes

Note: the specified resolution will take effect on the next mode set

SeeAlso: AH=00h

-----V-1012--BL31-----

INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (VGA, MCGA) - PALETTE LOADING

AH = 12h

BL = 31h

AL = new state

00h enable default palette loading

01h disable default palette loading

Return: AL = 12h if function supported

Desc: specify whether a default palette should be loaded when the display

```
mode is set
SeeAlso: AH=00h
-----V-1012--BL32-----
INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (VGA, MCGA) - VIDEO ADDRESSING
  AH = 12h
  BL = 32h
  AL = new state
    00h enable video addressing
    01h disable video addressing
Return: AL = 12h if function supported
Desc: specify whether the CPU should have access to video memory and the
      display adapters I/O registers
-----V-1012--BL33-----
INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (VGA, MCGA) - GRAY-SCALE SUMMING
  AH = 12h
  BL = 33h
  AL = new state
    00h enable gray scale summing
    01h disable gray scale summing
Return: AL = 12h if function supported
Desc: specify whether or not colors should be converted to gray scale when
      palette or color registers are loaded
SeeAlso: AX=101Bh,AX=BF06h,AX=BF08h
-----V-1012--BL34-----
INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (VGA) - CURSOR EMULATION
  AH = 12h
  BL = 34h
  AL = new state
    00h enable alphanumeric cursor emulation
    01h disable alphanumeric cursor emulation
Return: AL = 12h if function supported
Desc: specify whether the BIOS should automatically remap cursor start/end
      according to the current character height in text modes
SeeAlso: AH=01h,AH=03h
-----V-1012--BL35-----
INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (PS) - DISPLAY-SWITCH INTERFACE
  AH = 12h
  BL = 35h
  AL = subfunction
    00h initial adapter video off
    01h initial planar video on
```

02h switch active video off
 03h switch inactive video on
 80h *UNDOCUMENTED* set system board video active flag

ES:DX -> buffer for 128 byte save area (if AL = 00h-03h)

Return: AL = 12h if function supported

Desc: switch between two video adapters which may otherwise have address conflicts

SeeAlso: AX=BF00h,AX=BF01h,INT 6D"VGA"

-----V-1012--BL36-----

INT 10 - VIDEO - ALTERNATE FUNCTION SELECT (PS, VGA) - VIDEO REFRESH CONTROL

AH = 12h

BL = 36h

AL = new state

00h enable refresh

01h disable refresh

Return: AL = 12h if function supported

Desc: specify whether the contents of video memory should be displayed on the screen; disabling refresh effectively blanks the screen

Note: when display refresh is disabled, the entire screen displays the color specified by the DAC color register 00h; thus to actually blank the screen, the application must first ensure that that register has been set to black

BUG: GEM/VIEWMAX ET3000 and ET4000 drivers dated 1988/6/23 do not correctly reenable refresh after the screen has been blanked due to inactivity, resulting in a total loss of video synchronization

SeeAlso: AX=BF05h

-----V-1012--BL37-----

INT 10 - VIDEO - XGA - MAINFRAME INTERACTIVE SUPPORT

AH = 12h

BL = 37h

AL = text attribute type (00h normal VGA, 01h mainframe type)

Return: AL = 12h if supported

SeeAlso: #00026,AH=08h,#00042

Bitfields for XGA mainframe type character attributes:

Bit(s) Description (Table 00026)

7 blink double rate (75% ON time) or use color 8 as background
 (see AX=1003h)

6 reverse video

5 underlined

4 left-most and right-most dots of underline area inverted

3 foreground intensity/character font select

2-0 foreground color

SeeAlso: #00014,#00036

-----V-1012--BL38-----

INT 10 - IBM SurePath BIOS - Officially "Private" Function

AH = 12h

BL = 38h

SeeAlso: AX=1118h,AH=12h/BL=39h

-----V-1012--BL39-----

INT 10 - IBM SurePath BIOS - Officially "Private" Function

AH = 12h

BL = 39h

SeeAlso: AX=1118h,AH=12h/BL=38h,AH=12h/BL=3Ah

-----V-1012--BL3A-----

INT 10 - IBM SurePath BIOS - Officially "Private" Function

AH = 12h

BL = 3Ah

SeeAlso: AX=1118h,AH=12h/BL=39h

-----V-1012--BL53-----

INT 10 - Tseng ET-4000 BIOS v8.00 (1991/06/07) - BUG

AH = 12h

BL = 53h

Note: due to an omitted end-of-list marker, this version of the BIOS will
crash the system on this function

-----V-1012--BL57-----

INT 10 - Tseng ET-4000 BIOS v8.00 (1991/06/07) - BUG

AH = 12h

BL = 57h

Note: due to an omitted end-of-list marker, this version of the BIOS will
crash the system on this function

-----V-1012--BH55-----

INT 10 - VIDEO - ALTERNATE FUNC SELECT (ATI,Tatung,Taxan) - ENHANCED FEATURES

AH = 12h

BH = 55h

BL = subfunction

00h disable enhanced features

01h enable enhanced features

02h get status

Return: AL = status flags (see #00027)

03h disable register trapping (CGA emulation)

04h enable register trapping

```

    05h program video mode
ES:BP -> video mode table (see #00028)
    06h get mode table
AL = video mode
Return: ES:BP -> table suitable for mode AL
        (and subfunc BL=05h) (see #00028)
    BP = FFFFh on error
    SI = 0000h (ATI Mach32)

```

Note: the ATI Mouse driver (MOUSE.COM/MOUSE.SYS) uses subfunction 06h to get the mode table for mode 03h, and then looks for a signature string beginning within 32 bytes of the offset specified by the word at ES:000Ch

SeeAlso: AH=FFh"Oak"

Index: installation check;ATI video adapters

Bitfields for status flags:

Bit(s) Description (Table 00027)

```

3  set if enhanced features enabled
7-5  monitor type
    000 PS/2 mono
    001 PS/2 color
    010 multi-sync
    011 Taxan 650 25kHz
    100 RGB
    101 mono
    110 EGA
    111 Compaq internal

```

Format of ATI VGA Wonder video mode table:

Offset Size Description (Table 00028)

```

00h BYTE number of columns
01h BYTE maximum row (number of rows - 1)
02h BYTE scan lines per row
03h WORD video buffer size in bytes
05h 4 BYTES values for Sequencer registers 1-4
09h BYTE value for Miscellaneous Output register
0Ah 25 BYTES values for CRTC registers 00h-18h
    00h horizontal total size (chars)
    01h horizontal displayed (chars)
    02h horizontal sync position (chars)
    03h horizontal sync width (chars)

```

```

04h vertical total size (char rows)
05h vertical total adjust (scan lines)
06h vertical displayed (char rows)
07h vertical sync position (char rows)
08h interlace mode
09h max scan line in row
0Ah cursor start scan line
0Bh cursor end scan line
0Ch screen memory start (high)
0Dh screen memory start (low)
0Eh cursor address (high)
0Fh cursor address (low)
10h light pen (high)
11h light pen (low)
23h 20 BYTES default palette (values for Attribute Controller regs 00h-13h)
37h 9 BYTES values for Graphics Controller registers 00h-08h

```

-----V-1012--BL74-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG

AH = 12h

BL = 74h

Note: due to an omitted end-of-list marker, these versions of the BIOS will crash the system on this function

-----V-1012--BL80-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG

AH = 12h

BL = 80h

Note: due to an omitted end-of-list marker, these versions of the BIOS will crash the system on this function

-----V-1012--BL80-----

INT 10 - Cirrus Logic BIOS - INQUIRE VGA TYPE

AH = 12h

BL = 80h

Return: AX = controller type in bits 13-0 (see #00029)

bit 14: ???

bit 15: ???

BL = silicon revision number (bit 7 set if not available)

BH = ???

bit 2 set if using CL-GD 6340 LCD interface

SeeAlso: AH=12h/BL=81h,AH=12h/BL=82h,AH=12h/BL=85h,AH=12h/BL=9Ah,AH=12h/BL=A1h

(Table 00029)

Values for Cirrus Logic video controller type:

0000h no extended alternate select support
0001h reserved
0002h CL-GD510/520
0003h CL-GD610/620
0004h CL-GD5320
0005h CL-GD6410
0006h CL-GD5410
0007h CL-GD6420
0008h CL-GD6412
0010h CL-GD5401
0011h CL-GD5402
0012h CL-GD5420
0013h CL-GD5422
0014h CL-GD5424
0015h CL-GD5426
0016h CL-GD5420r1
0017h CL-GD5402r1
0018h CL-GD5428
0019h CL-GD5429
0020h CL-GD6205/15/25
0021h CL-GD6215
0022h CL-GD6225
0023h CL-GD6235
0024h CL-GD6245
0030h CL-GD5432
0031h CL-GD5434
0032h CL-GD5430
0033h CL-GD5434 rev. E and F
0035h CL-GD5440
0036h CL-GD5436
0039h CL-GD5446
0040h CL-GD6440
0041h CL-GD7542 (Nordic)
0042h CL-GD7543 (Viking)
0043h CL-GD7541 (Nordic Lite)
0050h CL-GD5452 (Northstar)
0052h CL-GD5452 (Northstar) ???

SeeAlso: #00732, #00743

-----V-1012--BL81-----

INT 10 - Cirrus Logic BIOS - GET BIOS VERSION NUMBER

AH = 12h

BL = 81h

Return: AH = BIOS major version

AL = BIOS minor version

SeeAlso: AH=12h/BL=80h,AH=12h/BL=82h

-----V-1012--BL82-----

INT 10 - Cirrus Logic BIOS - GET DESIGN REVISION CODE

AH = 12h

BL = 82h

Return: AL = chip revision

AH = ??? (AFh for v1.01)

SeeAlso: AH=12h/BL=80h,AH=12h/BL=81h,#00732 at INT 1A/AX=B102h

-----V-1012--BL84-----

INT 10 - Cirrus Logic BIOS v3.02 - INQUIRE OPTIONS

AH = 12h

BL = 84h

Return: AX = user options word (see #00030)

SeeAlso: AH=12h/BL=89h,AH=12h/BL=8Ah,AH=12h/BL=8Bh,AH=12h/BL=8Ch,AH=12h/BL=8Fh

SeeAlso: AH=12h/BL=90h,AH=12h/BL=9Ah

Bitfields for Cirrus Logic user options word:

Bit(s) Description (Table 00030)

1,0 centering

00 vertical centered, 01 from bottom, 10 from top, 11 reserved

3,2 720-dot fix-up options

00 OR every 8th and 9th pixel

01 display MGA mode from left

10 display MGA mode from right

11 skip every 9th pixel

4 ???

7-5 video mode (001 CGA, 010 MGA, 011 EGA, 100 VGA)

8 external monitor enabled instead of LCD panel

9 vertical expand mode enabled

10 8-bit mode instead of 16-bit mode

11 normal video rather than reverse video (for LCD)

12 attribute automap rather than attribute emulation

13 bold mode disabled (default)

14 fast bandwidth

15 ???

-----V-1012--BL85-----

INT 10 - Cirrus Logic BIOS - GET INSTALLED MEMORY

AH = 12h

BL = 85h

Return: AL = number of 64K banks of video memory

SeeAlso: AH=12h/BL=80h,AH=12h/BL=93h

-----V-1012--BL89-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - EN/DISABLE REVERSE VIDEO MODE

AH = 12h

BL = 89h

AL = new state (00h enabled, 01h disabled)

Return: nothing

SeeAlso: AH=12h/BL=84h,AH=12h/BL=8Ah,AH=12h/BL=8Bh

-----V-1012--BL8A-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - SET FRAME COLOR

AH = 12h

BL = 8Ah

AL = new gray-scale color (00h = black to 0Fh = white)

Return: nothing

SeeAlso: AH=12h/BL=84h,AH=12h/BL=89h,AH=12h/BL=8Fh

-----V-1012--BL8B-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - ENABLE/DISABLE BOLD MODE

AH = 12h

BL = 8Bh

AL = new state (00h enabled, 01h disabled)

Return: nothing

SeeAlso: AH=12h/BL=84h,AH=12h/BL=89h,AH=12h/BL=8Ch,AH=12h/BL=8Fh

-----V-1012--BL8C-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - SET AUTOMAP/EMULATE ATTRIBUTES

AH = 12h

BL = 8Ch

AL = new state

00h enable automap

01h disable automap and emulate attributes

Return: nothing

SeeAlso: AH=12h/BL=84h,AH=12h/BL=8Bh

-----V-1012--BL8F-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - ENABLE/DISABLE EXPAND MODE

AH = 12h

BL = 8Fh

AL = new state (00h enabled, 01h disabled)

Return: nothing

Note: when expand mode is enabled, the vertical dimension is enlarged to

full screen

SeeAlso: AH=12h/BL=84h,AH=12h/BL=89h,AH=12h/BL=8Bh,AH=12h/BL=90h

-----V-1012--BL90-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - SET CENTERING MODE

AH = 12h

BL = 90h

AL = new position

00h centered

01h from top

02h from bottom

03h from top

Return: nothing

SeeAlso: AH=12h/BL=84h,AH=12h/BL=8Fh

-----V-1012--BL91-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - SET 720-DOT FIXUP MODE

AH = 12h

BL = 91h

AL = new mode

00h display MGA mode from left of screen (default)

01h display MGA from right

02h skip every ninth pixel

03h OR every 8th and 9th pixel

Return: nothing

SeeAlso: AH=12h/BL=84h,AH=12h/BL=8Ah,AH=12h/BL=90h

-----V-1012--BL92-----

INT 10 - Cirrus Logic BIOS v3.02 - LCD panel - SWITCH DISPLAY

AH = 12h

BL = 92h

AL = new display (00h LCD, 01h external monitor)

Return: nothing

Note: the deselected display is disabled

SeeAlso: AH=12h/BL=84h,AH=12h/BL=89h,AH=12h/BL=90h,AH=12h/BL=94h

-----V-1012--BL93-----

INT 10 - Cirrus Logic BIOS - FORCE 8-BIT OR 16-BIT OPERATION

AH = 12h

BL = 93h

AL = new I/O width (00h = 16 bits, 01h = 8 bits)

Return: nothing

SeeAlso: AH=12h/BL=9Ah

-----V-1012--BL94-----

INT 10 - Cirrus Logic BIOS v3.02 - POWER CONSERVATION

AH = 12h

BL = 94h

AL = new state (00h wake up monitor, 01h shut down display)

Return: nothing

Note: AL=01h is reported not to work properly on the LCD panel

SeeAlso: AH=12h/BL=84h,AH=12h/BL=92h

-----V-1012--BL9A-----

INT 10 - Cirrus Logic BIOS - GET USER OPTIONS

AH = 12h

BL = 9Ah

Return: AX = options word 1 (see #00031)

CX = options word 2 (see #00032)

BX,DX reserved

SeeAlso: AH=12h/BL=80h,AH=12h/BL=84h,AH=12h/BL=93h,AH=12h/BL=A0h

SeeAlso: AH=12h/BL=A3h,AH=12h/BL=A4h,#00732 at INT 1A/AX=B102h

Bitfields for Cirrus Logic options word 1:

Bit(s) Description (Table 00031)

0,1 reserved

2-4 monitor type

5-6 maximum vertical resolution

7-9 reserved

10 force 8-bit operation

11-13 reserved

14 vertical refresh frequency at 640x480

=0 60 Hz

=1 high refresh

15 reserved

SeeAlso: AH=12h/BL=AEh

Bitfields for Cirrus Logic options word 2:

Bit(s) Description (Table 00032)

15-13 vertical refresh frequency at 1024x768

12-11 vertical refresh frequency at 800x600

10-6 reserved

5-4 vertical refresh frequency at 1280x1024

3-0 reserved

SeeAlso: #00033

Bitfields for Cirrus Logic Options word 2 (alternate???):

Bit(s) Description (Table 00033)


```
15-12 vertical refresh frequency at 1024x768
 0000 43i Hz
 0001 60 Hz
 0010 70 Hz
 0011 72 Hz
 0100 76 Hz
 other reserved
11-8 vertical refresh frequency at 800x600
 0000 56 Hz
 0001 60 Hz
 0010 72 Hz
 0011 75 Hz
 other reserved
7-4 maximum vertical resolution
 0000 480 scanlines
 0001 600 scanlines
 0010 768 scanlines
 0011 1024 scanlines
 other reserved
3-1 vertical refresh frequency at 1280x1024
 000 43i Hz
 001 60 Hz
 010 71.2 Hz (CL-GD5434 rev. E and F)
 011 74 Hz (CL-GD5434 rev. E and F)
 other reserved
0 reserved
```

SeeAlso: #00032

-----V-1012--BLA0-----

INT 10 - Cirrus Logic BIOS - GET VIDEO MODE AVAILABILITY

AH = 12h

BL = A0h

AL = video mode number (00h-7Fh)

Return: AH bit 0: video mode supported

BX = offset of BIOS subroutine to fixup standard video parameters

(call subroutine with DS:SI and ES:DI as returned by this call)

DS:SI -> standard video parameters or FFFFh:FFFFh

ES:DI -> supplemental video parameters or FFFFh:FFFFh

SeeAlso: AH=00h,AH=12h/BL=9Ah,AH=12h/BL=A1h

-----V-1012--BLA1-----

INT 10 - Cirrus Logic BIOS - READ MONITOR TYPE AND ID FROM 15-PIN CONNECTOR

AH = 12h

BL = A1h

Return: BH = monitor ID (see #00034)

BL = monitor type (00h color, 01h grayscale, 02h no display)

SeeAlso: AH=12h/BL=9Ah,AH=12h/BL=A2h

(Table 00034)

Values for Cirrus Logic monitor ID:

00h-08h reserved

09h IBM 8604/8507 or equivalent

0Ah IBM 8514 or equivalent

0Bh IBM 8515 or equivalent

0Dh IBM 8503 or equivalent

0Eh IBM 8512/8513 or equivalent

0Fh no monitor

-----V-1012--BLA2-----

INT 10 - Cirrus Logic BIOS - SET MONITOR HORIZONTAL RETRACE FREQUENCY

AH = 12h

BL = A2h

AL = retrace rate

00h standard VGA (31.5 kHz)

01h 8514-compatible (31.5 kHz and 35.5 kHz interlaced)

02h SuperVGA (31.5-35.1 kHz)

03h extended SuperVGA (31.5-35.5 kHz)

04h multi-frequency (31.5-37.8 kHz)

05h extended multi-frequency (31.5-48.0 kHz)

06h super multi-frequency (31.5-56.0 kHz)

07h extended super multi-frequency (31.5-64.0 kHz)

Return: nothing

SeeAlso: AH=12h/BL=9Ah,AH=12h/BL=A0h,AH=12h/BL=A4h

-----V-1012--BLA3-----

INT 10 - Cirrus Logic BIOS - SET VGA REFRESH

AH = 12h

BL = A3h

AL = refresh rate for 640x480 (00h normal, 01h high)

Return: nothing

SeeAlso: AH=12h/BL=9Ah,AH=12h/BL=A4h

-----V-1012--BLA4-----

INT 10 - Cirrus Logic BIOS - SET MONITOR TYPE

AH = 12h

BL = A4h

AL = options 1

bits 7-4 = vertical refresh at 640x480
(0 = 60Hz, 1 = 72Hz, other reserved)
bits 3-0 = maximum vertical resolution
(0 = 480 scanlines, 1 = 600, 2 = 768, 3 = 1024, other reserved)

BH = options 2

bits 7-4 = vertical refresh at 1024x768
(0=87Hz-int, 1=60Hz, 2=70Hz, 3=72Hz, 4=76Hz, other reserved)
bits 3-0 = vertical refresh at 800x600
(0 = 56Hz, 1 = 60Hz, 2 = 72Hz, other reserved)

CH = options 3

bits 7-4 = vertical refresh at 1280x1024
(0=87Hz-int, 1=60Hz, 2=70Hz, other reserved)
bits 3-0: reserved

CL,DX reserved

Return: nothing

SeeAlso: AH=12h/BL=9Ah,AH=12h/BL=A3h

-----V-1012--BLAE-----

INT 10 - Cirrus Logic BIOS - GET HIGH REFRESH

AH = 12h

BL = AEh

Return: AL = result

bits 7-1: reserved
bit 0: 640x480 high refresh rate
=0 72 Hz
=1 75 Hz

SeeAlso: AH=12h/BL=9Ah,AH=12h/BL=A3h,AH=12h/BL=A4h,AH=12h/BL=AFh

-----V-1012--BLAF-----

INT 10 - Cirrus Logic BIOS - SET HIGH REFRESH

AH = 12h

BL = AFh

AL = option

bits 7-1: reserved
bit 0: 640x480 high refresh rate
=0 72 Hz
=1 75 Hz

Return: nothing

SeeAlso: AH=12h/BL=9Ah,AH=12h/BL=A3h,AH=12h/BL=A4h,AH=12h/BL=AEh

-----V-1012--BLBE-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG

AH = 12h

BL = BEh

Note: due to an omitted end-of-list marker, these versions of the BIOS will

crash the system on this function

-----V-1012--BLC6-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG

AH = 12h

BL = C6h

Note: due to an omitted end-of-list marker, these versions of the BIOS will

crash the system on this function

-----V-1012--BLF0-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG

AH = 12h

BL = F0h

Note: due to an omitted end-of-list marker, these versions of the BIOS will

crash the system on this function

-----V-1012--BLF1-----

INT 10 - Tseng ET-4000 BIOS - GET/SET SCREEN REFRESH RATE

AH = 12h

BL = F1h

AL = subfunction

00h set refresh rate

01h get refresh rate

BH = video mode

00h 640x480

01h 800x600

02h 1024x768

03h 1280x1024

CX = new refresh rate (see #00035) if AL = 00h

Return: AL = 12h if supported

CX = current rate (for AL=00h, a changed CX indicates failure)

(Table 00035)

Values for Tseng ET4000 refresh rate:

CX 640x480 800x600 1024x768/1280x1024

00h 60 Hz 56 Hz interlaced

01h 72 Hz 60 Hz 60 Hz

02h 75 Hz 72 Hz 70 Hz

03h 90 Hz 75 Hz 75 Hz

04h -- 90 Hz --

-----V-1013-----

INT 10 - VIDEO - WRITE STRING (AT and later,EGA)

AH = 13h

AL = write mode
 bit 0: update cursor after writing
 bit 1: string contains alternating characters and attributes
 bits 2-7: reserved (0)

BH = page number

BL = attribute if string contains only characters

CX = number of characters in string

DH,DL = row,column at which to start writing

ES:BP -> string to write

Return: nothing

Notes: recognizes CR, LF, BS, and bell; for the ET4000 BIOS, scrolling,
 backspace, and CR only take place in the active page
 also available PC or XT with EGA or higher
 HP 95LX only supports write mode 00h
 IBM documents AL=10h,11h,20h,21h as "private" rather than "reserved"
 with PhysTechSoft's PTS ROM-DOS the AL,BH,BL,DH, and DL values are
 ignored on entry.

BUG: on the IBM VGA Adapter, any scrolling which may occur is performed on
 the active page rather than the requested page

SeeAlso: AH=09h,AH=0Ah,AH=13h"DOS/V"

-----J-1013-----

INT 10 - DOS/V - READ/WRITE DOUBLE-BYTE CHARACTER SET CHARACTERS/ATTRIBUTES

AH = 13h

AL = function

 10h read characters and standard attributes

 11h read characters and extended attributes

 12h write characters and standard attributes

 13h write characters and extended attributes

BH = 00h

CX = number of characters to transfer

DH,DL = row,column at which to start transfer

ES:BP -> buffer for/containing characters and attributes (see #00036)

Return: ES:BP buffer filled if reading

Program: DOS/V is a Japanese-language version of MS-DOS which can run on
 standard (non-Japanese) ATs and compatible equipped with a VGA or
 newer video adapter

Notes: subfunctions 11h and 13h are only valid when DOS/V is using video
 mode 73h

 the cursor position is not changed by these functions

 extended attributes consist of three bytes; the first is the standard
 character attribute byte, the second is as described below, and the

third is always zero in current versions

SeeAlso: AH=00h,AH=13h"VIDEO"

Bitfields for DOS/V second extended attribute byte:

Bit(s) Description (Table 00036)

7 underline using foreground color

6 reverse foreground/background specified in standard attribute byte

5-4 unused

3 vertical white grid line in cell

2 horizontal white grid line in cell

1-0 unused

SeeAlso: #00014,#00026

-----V-101400-----

INT 10 - VIDEO - LOAD USER-SPECIFIED LCD CHARACTER FONT (CONV,Compaq Port 386)

AX = 1400h

ES:DI -> character font

BH = number of bytes per character

08h or 10h (Compaq)

BL = 00h load main font (block 0)

01h load alternate font (block 1)

CX = number of characters to store

DX = character offset into RAM font area

Return: nothing

SeeAlso: AX=1100h,AX=1110h,AX=1401h

-----V-101401-----

INT 10 - VIDEO - LOAD SYSTEM ROM DEFAULT LCD CHARACTER FONT (CONV,CP386)

AX = 1401h

BL = font to load

00h main font (block 0)

01h alternate font (block 1)

Return: nothing

SeeAlso: AX=1100h,AX=1102h,AX=1400h

-----V-101402-----

INT 10 - VIDEO - SET MAPPING OF LCD HIGH INTENSITY ATTRIBUTES (CONV,CP386)

AX = 1402h

BL = subfunction

00h ignore high intensity attribute

01h map high intensity to reverse video

02h map high intensity to underscore

03h map high intensity to selected alternate font

B0h half intensity (Compaq)

B1h toggle active intensity bit interpretation (CP386)

Return: nothing

-----V-1015-----

INT 10 - VIDEO - GET PHYSICAL DISPLAY PARAMETERS (CONVERTIBLE)

AH = 15h

Return: AX = alternate display adapter type (see #00037)

ES:DI -> parameter table (see #00038)

SeeAlso: AH=1Bh

(Table 00037)

Values for PC Convertible alternate display adapter type:

0000h none

5140h LCD

5153h CGA

5151h mono

Format of PC Convertible display parameter table:

Offset Size Description (Table 00038)

00h WORD monitor model number

02h WORD vertical pixels per meter

04h WORD horizontal pixels per meter

06h WORD total vertical pixels

08h WORD total horizontal pixels

0Ah WORD horizontal pixel separation in micrometers

(width, center-to-center)

0Ch WORD vertical pixel separation in micrometers

(height, center-to-center)

-----V-1015-----

INT 10 - VIDEO - SET SUPERIMPOSE MODE (Sperry PC)

AH = 15h

AL = superimpose mode

00h show graphics screen

01h show text screen

02h show text screen superimposed on graphics screen

Return: nothing

-----J-1018-----

INT 10 - VIDEO - DOS/V - GET/SET FONT PATTERN

AH = 18h

AL = subfunction

00h get font pattern

01h set font pattern

BX = 0000h
CL = character size in bytes (01h,02h)
CH = 00h
DH = character width in pixels
DL = character height in pixels
ES:DI -> buffer for/containing font image
Return: AL = status (00h successful, else error)
ES:DI buffer filled for function 00h if successful
Note: the supported font sizes are 8x16 single-byte, 8x19 single-byte,
16x16 double-byte, and 24x24 double-byte
SeeAlso: AH=19h,INT 16/AH=14h

-----V-1019-----

INT 10 - Japanese VIDEO - DOUBLE-BYTE CHARACTER SET SHIFT INFORMATION

AH = 19h

???

Return: ???

SeeAlso: AH=18h,INT 16/AH=14h

-----V-101A00-----

INT 10 - VIDEO - GET DISPLAY COMBINATION CODE (PS,VGA/MCGA)

AX = 1A00h

Return: AL = 1Ah if function was supported

BL = active display code (see #00039)

BH = alternate display code (see #00039)

Notes: this function is commonly used to check for the presence of a VGA
this function is supported on the ATI EGA Wonder with certain
undocumented configuration switch settings, even though the EGA
Wonder does not support VGA graphics; to distinguish this case,
call AX=1C00h with CX nonzero, which the EGA Wonder does not support

SeeAlso: AH=12h/BL=35h,AX=1A01h,AH=1Bh,AH=1Ch,MEM 0040h:008Ah

Index: installation check;VGA

(Table 00039)

Values for display combination code:

00h no display
01h monochrome adapter w/ monochrome display
02h CGA w/ color display
03h reserved
04h EGA w/ color display
05h EGA w/ monochrome display
06h PGA w/ color display
07h VGA w/ monochrome analog display


```

08h  VGA w/ color analog display
09h  reserved
0Ah  MCGA w/ digital color display
0Bh  MCGA w/ monochrome analog display
0Ch  MCGA w/ color analog display
FFh  unknown display type

```

```
-----V-101A01-----
```

```
INT 10 - VIDEO - SET DISPLAY COMBINATION CODE (PS,VGA/MCGA)
```

```
AX = 1A01h
```

```
BL = active display code (see #00039)
```

```
BH = alternate display code
```

```
Return: AL = 1Ah if function was supported
```

```
SeeAlso: AH=12h/BL=35h,AX=1A00h
```

```
-----V-101B-----
```

```
INT 10 - VIDEO - FUNCTIONALITY/STATE INFORMATION (PS,VGA/MCGA)
```

```
AH = 1Bh
```

```
BX = implementation type
```

```
0000h return functionality/state information
```

```
ES:DI -> 64-byte buffer for state information (see #00040)
```

```
Return: AL = 1Bh if function supported
```

```
ES:DI buffer filled with state information
```

```
BUG: Trident 8900 (BIOS D3.0 1991/11/12) and Trident 9000 (C3.0 1991/10/25)
```

```
do not correctly set the fields at offsets 27h and 29h of the
state information
```

```
SeeAlso: AH=15h,AX=1A00h,AX=1F01h
```

```
Format of MCGA+ state information:
```

```
Offset Size Description (Table 00040)
```

```
00h  DWORD address of static functionality table (see #00045)
```

```
04h  BYTE video mode in effect
```

```
05h  WORD number of columns
```

```
07h  WORD length of regen buffer in bytes
```

```
09h  WORD starting address of regen buffer
```

```
0Bh  WORD cursor position for page 0
```

```
0Dh  WORD cursor position for page 1
```

```
0Fh  WORD cursor position for page 2
```

```
11h  WORD cursor position for page 3
```

```
13h  WORD cursor position for page 4
```

```
15h  WORD cursor position for page 5
```

```
17h  WORD cursor position for page 6
```

```
19h  WORD cursor position for page 7
```

1Bh WORD cursor "type" (start/stop scan lines)
1Dh BYTE active display page
1Eh WORD CRTC port address
20h BYTE current setting of PORT 03x8h
21h BYTE current setting of PORT 03x9h
22h BYTE number of rows - 1
23h WORD bytes/character
25h BYTE display combination code of active display
26h BYTE DCC of alternate display
27h WORD number of colors supported in current mode (0000h = mono)
29h BYTE number of pages supported in current mode
2Ah BYTE number of scan lines active
 (0,1,2,3) = (200,350,400,480)
 Tseng ET3000: (4,5,6 = 512,600,768)
2Bh BYTE primary character block
2Ch BYTE secondary character block
2Dh BYTE miscellaneous flags (see #00041)
2Eh BYTE non-VGA mode support (see #00042)
2Fh 2 BYTES reserved (00h)
31h BYTE video memory available
 00h = 64K, 01h = 128K, 02h = 192K, 03h = 256K
32h BYTE save pointer state flags (see #00043)
33h BYTE display information and status (see #00044)
34h 12 BYTES reserved (00h)

Bitfields for miscellaneous flags:

Bit(s) Description (Table 00041)

- 0 all modes on all displays on
- 1 gray summing on
- 2 monochrome display attached
- 3 default palette loading disabled
- 4 cursor emulation enabled
- 5 0 = intensity; 1 = blinking
- 6 flat-panel display is active
- 7 unused (0)

SeeAlso: #00040

Bitfields for non-VGA mode support:

Bit(s) Description (Table 00042)

- 7-5 reserved
- 4 132-column mode supported

3 =1 MFI attributes enabled (see AH=12h/BL=37h)
=0 VGA attributes
2 16-bit VGA graphics present
1 adapter interface driver required
0 BIOS supports information return for adapter interface

SeeAlso: #00040

Bitfields for save pointer state flags:

Bit(s) Description (Table 00043)

0 512 character set active
1 dynamic save area present
2 alpha font override active
3 graphics font override active
4 palette override active
5 DCC override active
6-7 unused (0)

SeeAlso: #00040

Bitfields for display information and status:

Bit(s) Description (Table 00044)

7 640x480 flat-panel can be used simultaneously with CRT controller
6-3 reserved
2 color display
1 flat-panel display active
0 flat-panel display attached

SeeAlso: #00040

Format of Static Functionality Table:

Offset Size Description (Table 00045)

00h BYTE modes supported 1
bit 0 to bit 7 = 1 modes 0,1,2,3,4,5,6,7 supported
01h BYTE modes supported 2
bit 0 to bit 7 = 1 modes 8,9,0Ah,0Bh,0Ch,0Dh,0Eh,0Fh supported
02h BYTE modes supported 3
bit 0 to bit 3 = 1 modes 10h,11h,12h,13h supported
bit 4 to bit 7 unused (0)
03h BYTE (IBM) reserved
(Tseng ET3000/4000) modes supported 4
bit 0 to bit 7 = modes 18h-1Fh supported
04h BYTE (IBM) reserved
(Tseng ET3000/4000) modes supported 5

bit 0 to bit 7 = modes 20h-27h supported
05h BYTE (IBM) reserved
 (Tseng ET3000/4000) modes supported 6
 bit 0 to bit 7 = modes 28h-2Fh supported
06h BYTE (IBM) reserved
 (Tseng ET3000/4000) modes supported 7
 bit 0 to bit 7 = modes 30h-37h supported
07h BYTE scan lines supported
 bit 0 to bit 2 = 1 if scan lines 200,350,400 supported
 bits 3-7 = unused (0)
08h BYTE total number of character blocks available in text modes
09h BYTE maximum number of active character blocks in text modes
0Ah WORD miscellaneous function support flags (see #00046)
0Ch WORD reserved
0Eh BYTE save pointer function flags (see #00047)
0Fh BYTE reserved

SeeAlso: #00040

Bitfields for miscellaneous function support flags:

Bit(s) Description (Table 00046)

- 0 all modes on all displays function supported
- 1 gray summing function supported
- 2 character font loading function supported
- 3 default palette loading enable/disable supported
- 4 cursor emulation function supported
- 5 EGA palette present
- 6 color palette present
- 7 color-register paging function supported
- 8 light pen supported (see AH=04h)
- 9 save/restore state function 1Ch supported
- 10 intensity/blinking function supported (see AX=1003h)
- 11 Display Combination Code supported (see #00039)
- 12-15 unused (0)

SeeAlso: #00040

Bitfields for save pointer function flags:

Bit(s) Description (Table 00047)

- 0 512 character set supported
- 1 dynamic save area supported
- 2 alpha font override supported
- 3 graphics font override supported

4 palette override supported

5 DCC extension supported

6-7 unused (0)

SeeAlso: #00040

-----V-101C-----

INT 10 - VIDEO - SAVE/RESTORE VIDEO STATE (PS50+,VGA)

AH = 1Ch

AL = function

00h return state buffer size

Return: BX = number of 64-byte blocks needed

01h save video state

ES:BX -> buffer

02h restore video state

ES:BX -> buffer containing previously saved state

CX = requested states (see #00048)

Return: AL = 1Ch if function supported

Notes: many BIOSes corrupt the video registers when saving the state, so a program should restore the state immediately after saving it (the saved data is uncorrupted)

the BIOS data area consists of the 96 bytes from 0040h:0049h-00A8h

this function is not supported when DOS/V is running

this function may be a better VGA installation check than the usual

INT 10/AX=1A00h, since some late-model EGA cards (such as the ATI

EGA Wonder) supported that call

SeeAlso: AX=1A00h,AX=5F90h,AX=5FA0h,MEM 0040h:00A8h

Bitfields for requested states:

Bit(s) Description (Table 00048)

0 video hardware (see #00049)

1 BIOS data areas

2 color registers and DAC state (see #00050)

3-15 unused (0)

SeeAlso: #00186

Format of VGA video hardware state:

Offset Size Description (Table 00049)

00h BYTE sequencer index register

01h BYTE CRTC index register

02h BYTE graphics controller index register

03h BYTE attribute controller index register

04h BYTE feature controller register

```

05h  4 BYTES sequencer registers
09h  BYTE  sequencer register 0
0Ah  25 BYTES CRTC registers 0-8
23h  16 BYTES palette registers 00h-0Fh
33h  4 BYTES attribute registers 10h-13h
37h  9 BYTES graphics controller registers 0-8
40h  BYTE  CRTC base address (low)
41h  BYTE  CRTC base address (high)
42h  BYTE  plane 0 latch
43h  BYTE  plane 1 latch
44h  BYTE  plane 2 latch
45h  BYTE  plane 3 latch

```

Format of VGA DAC state:

Offset Size Description (Table 00050)

```

00h  BYTE  read/write mode DAC
01h  BYTE  pixel address
02h  BYTE  pixel mask
03h  768 BYTES  color data (256 triples)
303h  BYTE  color select register

```

-----J-101D-----

INT 10 - VIDEO - DOS/V - SHIFT STATUS LINE CONTROL

AH = 1Dh

AL = function

00h enable shift status line(s)

BX = number of lines to reserve at bottom of screen (usu. 1)

01h disable shift status line

BX = number of lines reserved at bottom of screen (usually 1)

02h get number of status lines

Return: BX = number of lines reserved for shift status

SeeAlso: AH=19h

-----V-101D-----

INT 10 - SpeedSTAR Plus BIOS v4.23+ - SET SYNC PARAMETERS

AH = 1Dh

AL = video mode

ES = caller's segment

Return: nothing

Note: the caller's segment contains at offset 5Ch (FCB field in PSP)

or 100h a table with sync parameters (see #00051) (BIOS looks at both offsets)

Format of SpeedSTAR Plus sync table:

Offset	Size	Description (Table 00051)
00h	9 BYTES	ID contains string 'ey5CENTER'
09h	5 BYTES	sync parameters for 640x480 modes 11h,12h,25h,26h,2Eh
0Bh	5 BYTES	sync parameters for 800x600 modes 29h,30h,2Ah
13h	5 BYTES	sync parameters for 1024x768 modes 37h,38h

-----V-101D-----

INT 10 - Tseng ET-4000 BIOS v3.00 and v8.00 - BUG
AH = 1Dh

Note: these versions of the BIOS jump to a random location on this function
due to a fencepost error

-----V-101DAABXFDEC-----

INT 10 U - Diamond Stealth64 Video - INSTALLATION CHECK
AX = 1DAAh
BX = FDECh

Return: BX = CDEFh if Diamond Stealth64 Video 2001-series video card installed
AL = number of megabytes of video memory
AH = card type??? (4Bh for VL-Bus/5Bh for PCI)
CX = ??? (0000h)
SI:DI -> signature/copyright string

-----V-101E00-----

INT 10 - VIDEO - FLAT-PANEL - READ INFORMATION
AX = 1E00h

Return: AL = 1Eh if function supported
BL = flat-panel status (see #00052)
ES:DI -> information table (see #00053)

Note: IBM classifies this function as optional

SeeAlso: AX=1E01h,AX=1E02h,AX=1E03h,AX=1E04h,AX=1E05h,AX=1EFEh

Bitfields for flat-panel status:

Bit(s)	Description (Table 00052)
7-3	reserved
2	color display
1	flat panel is active
0	flat panel is attached

SeeAlso: #00053

Format of flat-panel information table:

Offset	Size	Description (Table 00053)
00h	BYTE	number of WORDs in the remainder of the table
01h	WORD	number of vertical pixels per meter

03h WORD number of horizontal pixels per meter
 05h WORD total number of vertical pixels
 07h WORD total number of horizontal pixels
 09h WORD vertical pixel separation in micrometers (center-to-center)
 0Bh WORD horizontal pixel separation in micrometers
 0Dh WORD range of gray levels available, less 1
 0Fh WORD number of red levels available, less 1 (0000h if nonochrome)
 11h WORD number of green levels available, less 1 (0000h if nonochrome)
 13h WORD number of blue levels available, less 1 (0000h if nonochrome)

SeeAlso: #00052

-----V-101E01-----

INT 10 - VIDEO - FLAT-PANEL - LCD/CRT DISPLAY CONTROL

AX = 1E01h

BH = function

bit 7: =1 set display control, =0 query control

bits 6-0: reserved (0)

---if BH bit 7 set---

BL = new display combination (see #00054)

Return: AL = 1Eh if function supported

BH = results

bit 7: query/set (copied from input)

bits 6-3: reserved (0)

bit 2: simultaneous display is supported by hardware

bit 1: LCD/CRT display control supported by hardware

bit 0: set operation was successful (always clear on get)

BL = active display combination (see #00054)

SeeAlso: AX=1E00h,AX=1E02h

Bitfields for Flat-Panel display combination:

Bit(s) Description (Table 00054)

7-2 reserved (0)

1-0 combination

00 reserved

01 LCD active

10 CRT active

11 both LCD and CRT active (simultaneous display)

SeeAlso: #00055

-----V-101E02-----

INT 10 - VIDEO - FLAT-PANEL - DISPLAY POSITION

AX = 1E02h

BH = function


```
    bit 7: =1 set display position, =0 query position
    bits 6-0: reserved (0)
---if BH bit 7 set---
BL = new position setting (see #00055)
Return: AL = 1Eh if function supported
    BH = results
    bit 7: query/set (copied from input)
    bits 6-4: reserved (0)
    bit 3: positioning to top is supported by hardware
    bit 2: positioning to center is supported by hardware
    bit 1: positioning to bottom is supported by hardware
    bit 0: set operation was successful (always clear on get)
    BL = active position setting (see #00055)
SeeAlso: AX=1E00h,AX=1E01h
```

Bitfields for Flat-Panel position setting:

Bit(s) Description (Table 00055)

```
7-2 reserved (0)
1-0 position
  00 center
  01 top
  10 bottom
  11 reserved
```

SeeAlso: #00054,#00056

-----V-101E03-----

INT 10 - VIDEO - FLAT-PANEL - VERTICAL EXPANSION FOR TEXT/GRAPHICS MODES

```
AX = 1E03h
BH = function
    bit 7: =1 set vertical expansion, =0 query expansion
    bits 6-0: reserved (0)
---if BH bit 7 set---
BL = new vertical expansion setting (see #00056)
Return: AL = 1Eh if function supported
    BH = results
    bit 7: query/set (copied from input)
    bits 6-4: reserved (0)
    bit 3: vertical expansion of text is supported by hardware
    bit 2: vertical expansion of graphics is supported by hardware
    bit 1: hardware supports independent control of expansion in
           text and graphics modes
    bit 0: set operation was successful (always clear on get)
```

BL = active vertical expansion setting (see #00056)

SeeAlso: AX=1E00h,AX=1E02h,AX=1E04h

Bitfields for Flat-Panel vertical expansion setting:

Bit(s) Description (Table 00056)

7-2 reserved (0)

1 vertical expansion turned on for text modes

0 vertical expansion turned on for graphics modes

SeeAlso: #00054,#00055,#00057

-----V-101E04-----

INT 10 - VIDEO - FLAT-PANEL - NORMAL/REVERSE VIDEO FOR TEXT/GRAPHICS MODES

AX = 1E04h

BH = function

bit 7: =1 set normal/reverse video, =0 query normal/reverse

bits 6-0: reserved (0)

---if BH bit 7 set---

BL = new normal/reverse video setting (see #00057)

Return: AL = 1Eh if function supported

BH = results

bit 7: query/set (copied from input)

bits 6-4: reserved (0)

bit 3: hardware supports reverse video for text modes

bit 2: hardware supports reverse video for graphics modes

bit 1: hardware supports independent control of reverse video
in text and graphics modes

bit 0: set operation was successful (always clear on get)

BL = active normal/reverse video setting (see #00057)

SeeAlso: AX=1E00h,AX=1E03h,AX=1E05h

Bitfields for Flat-Panel normal/reverse video setting:

Bit(s) Description (Table 00057)

7-2 reserved (0)

1 reverse video for text

0 reverse video for graphics

SeeAlso: #00054,#00056,#00058,#00059

-----V-101E05-----

INT 10 - VIDEO - FLAT-PANEL - BRIGHTNESS CONTROL

AX = 1E05h

BH = function

bit 7: =1 set brightness, =0 query brightness

bits 6-0: reserved (0)

```
---if BH bit 7 set---
BL = new brightness setting (see #00058)
Return: AL = 1Eh if function supported
      BH = results
      bit 7: query/set (copied from input)
      bits 6-2: reserved (0)
      bit 1: software brightness control is supported
      bit 0: set operation was successful (always clear on get)
      BL = active brightness setting (see #00058)
Note: this function operates independently of AX=1E07h
SeeAlso: AX=1E00h,AX=1E04h,AX=1E06h,AX=1E07h
```

Bitfields for Flat-Panel brightness control:

Bit(s) Description (Table 00058)

7-1 reserved (0)

0 high brightness

SeeAlso: #00057,#00059

-----V-101E06-----

INT 10 - VIDEO - FLAT-PANEL - CONTRAST CONTROL FOR TEXT/GRAPHICS

AX = 1E06h

BH = function

bit 7: =1 set contrast control, =0 query contrast

bits 6-0: reserved (0)

---if BH bit 7 set---

BL = new normal/reverse video setting (see #00059)

Return: AL = 1Eh if function supported

BH = results

bit 7: query/set (copied from input)

bits 6-4: reserved (0)

bit 3: software contrast control supported for text

bit 2: software contrast control supported for graphics

bit 1: hardware supports independent control of contrast
in text and graphics modes

bit 0: set operation was successful (always clear on get)

BL = active contrast setting (see #00059)

Note: this function operates independently of AX=1E08h

SeeAlso: AX=1E00h,AX=1E04h,AX=1E05h,AX=1E07h,AX=1E08h

Bitfields for Flat-Panel contrast control:

Bit(s) Description (Table 00059)

7-2 reserved (0)

1 high contrast for text
0 high contrast for graphics

SeeAlso: #00057,#00058

-----V-101E07-----

INT 10 - VIDEO - FLAT-PANEL - BRIGHTNESS SETTING

AX = 1E07h

BH = function

bit 7: =1 set brightness control, =0 query brightness

bit 6: use standard brightness

bits 5-0: reserved (0)

---if BH bits 7,6=10---

BL = brightness (00h = minimum, FFh = maximum)

Return: AL = 1Eh if function supported

BH = results

bit 7: query/set (copied from input)

bit 6: standard/custom (copied from input)

bits 5-2: reserved (0)

bit 1: software brightness control is supported

bit 0: set operation was succesful (always clear on get)

BL = brightness (00h = minimum, FFh = maximum)

Note: this function operates independently of AX=1E05h

SeeAlso: AX=1E00h,AX=1E05h,AX=1E08h

-----V-101E08-----

INT 10 - VIDEO - FLAT-PANEL - CONTRAST SETTING

AX = 1E08h

BH = function

bit 7: =1 set contrast control, =0 query contrast

bit 6: use standard contrast

bits 5-0: reserved (0)

---if BH bits 7,6=10---

BL = contrast (00h = minimum, FFh = maximum)

Return: AL = 1Eh if function supported

BH = results

bit 7: query/set (copied from input)

bit 6: standard/custom (copied from input)

bits 5-2: reserved (0)

bit 1: software contrast control is supported

bit 0: set operation was succesful (always clear on get)

BL = contrast (00h = minimum, FFh = maximum)

Note: this function operates independently of AX=1E06h

SeeAlso: AX=1E00h,AX=1E06h,AX=1E07h

-----V-101EFE-----

INT 10 - VIDEO - FLAT-PANEL - FUNCTION SUPPORT FLAG

AX = 1EFEh

ES:DI -> DWORD buffer for bitmap of supported functions

(set to all zeros before calling)

Return: ES:DI buffer updated with mask of supported functions

Index: installation check;flat-panel display support

SeeAlso: AX=1E00h,AX=1E06h

-----V-101F00-----

INT 10 - VIDEO - XGA - GET DMQS (Display Mode Query and Set) DATA LENGTH

AX = 1F00h

Return: AL = 1Fh if supported

BX = number of bytes of DMQS data

Note: not supported on the original IBM XGA, only on XGA-NI (non-interlaced)

and later models.

SeeAlso: AX=1F01h

-----V-101F01-----

INT 10 - VIDEO - XGA - READ DMQS DATA

AX = 1F01h

ES:DI -> user buffer for return data (call AX=1F00h for size)

Return: AL = 1Fh if function supported

user buffer filled with DMQS data (see #00060)

Note: not supported on the original IBM XGA, only on XGA-NI (non-interlaced)

and later models.

SeeAlso: AH=1Bh,AX=1F00h,AX=3000h

Format of XGA DMQS buffer:

Offset Size Description (Table 00060)

00h WORD offset (in bytes) to DMQS data for next XGA instance

02h BYTE slot number

03h BYTE XGA implementation function level identifier

04h BYTE XGA implementation resolution level identifier

05h WORD vendor identifier - identifies card vendor

07h WORD vendor defined field

09h WORD XGA adapter I/O register base address

0Bh WORD XGA coprocessor register base address

(paragraph--multiply by 10h to get physical address)

0Dh WORD 1 Megabyte system video memory aperture

0000h if not allocated

(Multiply by 100000h to get physical address)

0Fh WORD 4 Megabyte system video memory aperture

0000h if not allocated
(multiply by 100000h to get physical address)
11h WORD video memory base address
(multiply by 100000h to get physical address)
13h WORD composite ID of the attached display
15h BYTE amount of video memory available, in multiples of 256K bytes
16h DWORD alternate XGA coprocessor register base address. 0 = none.
1Ah var DMQS Data for further XGA Instances (as above)

Note: "Instances" refers to the capability of having up to 8 XGA
adapters in one computer.

SeeAlso: #00061

-----V-101F02-----
INT 10 - VIDEO - XGA - GET SVGA DMQS DATA LENGTH
AX = 1F02h

Return: AL = 1Fh if function supported
BX = length of SVGA DMQS data in bytes

SeeAlso: AX=1F00h,AX=1F03h

-----V-101F03-----
INT 10 - VIDEO - XGA - GET SVGA DMQS DATA
AX = 1F03h
ES:DI -> buffer for SVGA DMQS data (see #00061)

Return: AL = 1Fh if function supported
ES:DI buffer filled

SeeAlso: AX=1F00h,AX=1F02h

Format of SVGA DMQS data:

Offset	Size	Description (Table 00061)
00h	WORD	offset to DMQS data for next SVGA instance or 0000h
02h	BYTE	reserved
03h	BYTE	SVGA implementation functional level
04h	BYTE	SVGA implementation resolution level
05h	WORD	vendor ID
07h	WORD	vendor-specific
09h	7 BYTES	reserved
10h	BYTE	SVGA linear address window size
00h		64K
01h		1M
02h		2M
03h		4M
11h	WORD	SVGA linear address window location or 0000h if not allocated (multiply by 10000h [shift left 16 bits] to get physical addr)

13h WORD attached monitor's composite monitor ID
15h BYTE available video RAM in multiples of 256K
16h 11 BYTES reserved

SeeAlso: #00060

-----V-103000CX0000-----

INT 10 - VIDEO - LOCATE 3270PC CONFIGURATION TABLE (INSTALLATION CHECK)

AX = 3000h

CX = 0000h

DX = 0000h

Return: CX:DX -> 3270PC configuration table (see #00062)

CX:DX = 0000h:0000h if 3270PC Control Program not active

SeeAlso: AX=1F01h

Format of 3270 PC configuration table:

Offset Size Description (Table 00062)

00h BYTE aspect ratio X

01h BYTE aspect ratio Y

02h BYTE monitor type (see #00063)

03h BYTE reserved

04h BYTE adapter ID

00h = 5151/5272 adapter

04h = 5151/5272 with XGA adapter

30h = 3295 or 3270PC G/GX adapter

05h BYTE reserved

06h BYTE function flags 1 (see #00064)

07h BYTE function flags 2

bit 6: GPI graphics supported

08h WORD segment address of Control Program Level table (see #00065)

0Ah 10 BYTES reserved

(Table 00063)

Values for 3270 PC monitor type:

00h 5151 (mono) or 5272 (color)

01h 3295

02h 5151 or 5272 with XGA (???) graphics adapter

03h 5279 with 3270PC G adapter

04h 5379 model C01 with 3270PC GX adapter

05h 5379 model M01 with 3270PC GX adapter

07h non-3270PC with 3270 Workstation Program

FFh 3270PC Control Program not loaded

Bitfields for 3270 PC function flags 1:

Bit(s) Description (Table 00064)

7 mono text, 1 page
 6 color text, 1 page
 5 color text, 4 pages
 4 CGA color graphics
 3 720x350 two-color graphics
 2 360x350 four-color graphics
 1 720x350 eight-color graphics

Format of Control Program Level table:

Offset Size Description (Table 00065)

00h WORD program version
 02xxh = 3270PC Control Program v2.xx
 03xxh = 3270PC Control Program v3.xx
 04xxh = 3270 Workstation Program v1.xx
 02h BYTE Control Program ID (00h)
 03h 27 BYTES Control Program Descriptor ("IBM 3270 PC CONTROL PROGRAM")

-----a-103800-----

INT 10 - Tinytalk Personal v1.09f+ - GET CONFIGURATION INFO

AX = 3800h

Return: ES:DI -> configuration info (see #00066)

Program: Tinytalk is a shareware screen reader by OMS Development/Eric Bohlman

Notes: this call is also used as the installation check by verifying that the returned ES:DI points at valid configuration info

Tinytalk v1.09f ignores AL and always returns the configuration info;

v1.10 adds several subfunctions selected with AL

SeeAlso: AX=3801h,AX=3803h,AX=3806h,AH=39h,INT 14/AX=F0F1h

SeeAlso: INT 2F/AX=FB00h"AutoBraille"

Index: installation check;Tinytalk Personal

Format of Tinytalk Personal configuration info:

Offset Size Description (Table 00066)

00h 8 BYTES signature "TTCONFIG"
 08h WORD size of configuration data, not counting signature, this WORD,
 or the following byte
 0Ah BYTE ???
 0Bh ??? configuration data

-----a-103801-----

INT 10 - Tinytalk Personal v1.10 - ???

AX = 3801h


```
DL = ???
Return: ???
SeeAlso: AX=3800h,AX=3802h
-----a-103802-----
INT 10 - Tinytalk Personal v1.10 - ???
  AX = 3802h
  DL = ???
Return: ???
SeeAlso: AX=3800h,AX=3801h
-----a-103803-----
INT 10 - Tinytalk Personal v1.10 - GET ???
  AX = 3803h
Return: AL = ???
SeeAlso: AX=3800h,AX=3804h,AX=3805h
-----a-103804-----
INT 10 - Tinytalk Personal v1.10 - GET ???
  AX = 3804h
Return: AX = ???
SeeAlso: AX=3800h,AX=3803h,AX=3805h
-----a-103805-----
INT 10 - Tinytalk Personal v1.10 - GET ???
  AX = 3805h
Return: AL = ???
SeeAlso: AX=3800h,AX=3803h,AX=3804h
-----a-103806-----
INT 10 - Tinytalk Personal v1.10 - ???
  AX = 3806h
  ES:DX -> ASCIZ ???
Return: ???
SeeAlso: AX=3800h
-----a-1039-----
INT 10 - Tinytalk Personal v1.10 - ???
  AH = 39h
Program: Tinytalk is a shareware screen reader by OMS Development/Eric Bohlman
Note: this function was a NOP in TTDEMO.EXE, but may be implemented in other
      variants of Tinytalk
-----V-1040-----
INT 10 - VIDEO - Hercules GRAFIX - "GMODE" - SET GRAPHICS MODE
  AH = 40h
Return: nothing
Program: Hercules GRAFIX (or GRAPH X, as it is called in some documentation)
```

provides text output in graphics mode and some graphics primitives
on the Hercules Graphics Card

Desc: switch the Hercules Graphics Card into graphics mode (720x348) without
clearing the screen; make Page 0 the default for both writing and
display, and set LEVEL (see AH=44h) to 01h

SeeAlso: AH=00h,AH=41h,AH=42h

Index: video modes;Hercules

-----V-1041-----

INT 10 - VIDEO - Hercules GRAFIX - "TMODE" - SET TEXT MODE

AH = 41h

Return: nothing

Desc: switch the Hercules Graphics Card into text mode

SeeAlso: AH=00h,AH=40h

Index: video modes;Hercules

-----V-1042-----

INT 10 - VIDEO - Hercules GRAFIX - "CLRSCR" - CLEAR CURRENT PAGE

AH = 42h

Return: nothing

SeeAlso: AH=45h

-----V-1043-----

INT 10 - VIDEO - Hercules GRAFIX - "GPAGE" - SELECT DRAWING PAGE

AH = 43h

AL = page number (0,1)

Return: nothing

Desc: specify which of the two pages of video memory is to be used for
output

SeeAlso: AH=05h,AH=44h,AH=45h

-----V-1044-----

INT 10 - VIDEO - Hercules GRAFIX - "LEVEL" - SELECT DRAWING FUNCTION

AH = 44h

AL = drawing function

00h clear pixels

01h set pixels

02h invert pixels

Return: nothing

Desc: specify how graphics output will change the display

SeeAlso: AH=40h,AH=45h,AH=46h,AH=4Bh,AH=4Ch,AH=4Dh

-----V-1045-----

INT 10 - VIDEO - Hercules GRAFIX - "DISP" - SELECT PAGE TO DISPLAY

AH = 45h

AL = page number (0,1)

Return: nothing

Desc: specify which of the two pages of video memory is visible on screen

SeeAlso: AH=05h,AH=42h,AH=43h

-----V-1046-----

INT 10 - VIDEO - Hercules GRAFIX - "PLOT" - DRAW ONE PIXEL

AH = 46h

DI = x (0-719)

BP = y (0-347)

Return: nothing

Notes: function 44h determines operation and function 43h which page to use
if the indicated coordinate is invalid, this function does nothing

SeeAlso: AH=0Ch,AH=47h,AH=49h,AH=4Ch,AH=4Dh

-----V-1047-----

INT 10 - VIDEO - Hercules GRAFIX - "GETPT" - FIND PIXEL VALUE

AH = 47h

DI = x (0-719)

BP = y (0-347)

Return: AL = 00h pixel clear

AL = 01h pixel set

Note: function 43h specifies which page is used

SeeAlso: AH=0Dh,AH=46h

-----V-1048-----

INT 10 - VIDEO - Hercules GRAFIX - "MOVE" - MOVE TO POINT

AH = 48h

DI = x (0-719)

BP = y (0-347)

Return: nothing

Desc: specify the location from which to start the next graphics output

SeeAlso: AH=49h

-----V-1049-----

INT 10 - VIDEO - Hercules GRAFIX - "DLINE" - DRAW TO POINT

AH = 49h

DI = x (0-719)

BP = y (0-347)

Return: nothing

Note: function 48h or 49h specify first point, 44h operation and 43h page to
use

SeeAlso: AH=43h,AH=44h,AH=48h,AH=4Ch,AH=4Dh

-----V-104A-----

INT 10 - VIDEO - Hercules GRAFIX - "BLKFIL" - BLOCK FILL

AH = 4Ah

DI = x coordinate of lower left corner
BP = y coordinate of lower left corner
BX = height in pixels
CX = width in pixels

Return: nothing

Desc: draw a solid rectangle of the specified size at the given location

SeeAlso: AH=4Eh

-----V-104B-----

INT 10 - VIDEO - Hercules GRAFIX - "TEXT" - DISPLAY CHARACTER

AH = 4Bh
AL = character to display
DI = x (0-719)
BP = y (0-347)

Return: nothing

Notes: unlike the other BIOS character functions character position is
specified in pixels rather than rows and columns
the character is writing using the drawing function last specified by
AH=44h

SeeAlso: AH=09h,AH=0Ah,AH=44h

-----P-104B-----

INT 10 - FRIEZE - API

AH = 4Bh
CL = function
00h (v6.x-) print window
AL = mode
00h character
01h normal
02h sideways
01h load window
ES:BX -> ASCIZ filename from which to read
02h save window
ES:BX -> ASCIZ filename to which to write
03h set print width
AL = width in 1/4 inches
04h set print height
AL = height in 1/4 inches
05h reserved
06h set left margin
AL = printout margin in 1/4 inches
07h set window size
ES:BX -> four-WORD structure with Xmin, Ymin, Xmax, Ymax

```
08h reserved
09h set patterns
ES:BX -> 16-BYTE vector of screen->printer color correspondnces
0Ah get patterns
ES:BX -> 16-BYTE buffer for color correspondences
0Bh set mode
AL = mode
0Ch (v7.41) ???
AL = ???
00h ??? (calls original INT 05)
else ???
0Dh (v7.41) ???
AL = ???
00h ??? (calls original INT 05)
else ???
0Eh (v7.41) ???
???
0Fh get window
ES:BX -> four-WORD buffer for Xmin, Ymin, Xmax, Ymax
10h set print options
ES:BX -> printer options in same format as FRIEZE cmdline
11h initialize
ES:BX -> three-WORD array from CARDS.DAT for HRes, VRes, code
12h (v7.41) ???
???
13h (v7.41) ???
???
14h get version
Return: AH = major version (00h if FRIEZE version before 7)
AL = minor version
15h set parameters
ES:BX -> parameter table (see #00068)
16h get parameters
ES:BX -> buffer for parameter table (see #00068)
17h get printer resolution
ES:BX -> 12-WORD table for six horizontal/vertical resol pairs
18h (v8.0 only) reserved
50h (v7.41) get ???
Return: AX = ???
51h (v7.41) get ???
Return: ES = ??? (seen 2348h)
```

AX = ??? (seen 8432h)

Return: AX = status (see #00067)

SeeAlso: AH=0Fh/SI=F123h

(Table 00067)

Values for FRIEZE function status:

00h successful
01h user aborted printout with ESC
02h reserved
03h file read error
04h file write error or printer error
05h file not found
06h invalid header (not an image or wrong screen mode) or can't create file
07h file close error
08h disk error
09h (v7.0+) printer error
0Ah invalid function
0Bh (v7.0+) can't create file
0Ch (v7.0+) wrong video mode

Format of FRIEZE parameter table:

Offset Size Description (Table 00068)

00h WORD top margin (1/100 inch)
02h WORD left margin (1/100 inch)
04h WORD horizontal size (1/100 inch)
06h WORD vertical size (1/100 inch)
08h WORD quality/draft mode
 00h draft mode
 01h quality mode
 02h use horizontal/vertical resolution for output resolution
0Ah WORD printer horizontal resolution (dots per inch)
0Ch WORD printer vertical resolution (dots per inch)
0Eh WORD reserved (FFFFh)

Note: any field which should remain unchanged may be filled with FFFFh

-----V-104C-----

INT 10 - VIDEO - Hercules GRAFIX - "ARC" - DRAW ARC

AH = 4Ch

AL = quadrant (1 = upper right, 2 = upper left, etc)

DI = x coordinate of center

BP = y coordinate of center

BX = radius

Return: nothing

SeeAlso: AH=49h,AH=4Dh

-----V-104D-----

INT 10 - VIDEO - Hercules GRAFIX - "CIRC" - DRAW CIRCLE

AH = 4Dh

DI = x of center

BP = y of center

BX = radius

Return: nothing

SeeAlso: AH=49h,AH=4Ch

-----V-104E-----

INT 10 - VIDEO - Hercules GRAFIX - "FILL" - FILL AREA

AH = 4Eh

DI = x coordinate of an interior point

BP = y coordinate of an interior point

Return: nothing

Desc: fill a convex polygonal area bounded by a contiguous line of the
opposite color with the border color

Note: the first fill makes the figure solid, the second erases it

SeeAlso: AH=4Ah

-----V-104E00-----

INT 10 - VESA XGA BIOS Extensions - GET XGA ENVIRONMENT INFORMATION

AX = 4E00h

ES:DI -> 256-byte buffer for XGA information (see #00069)

Return: AL = 4Eh if function supported

AH = status

00h successful

else error code

Desc: determine whether VESA XGA extensions are present and the capabilities
supported by the display adapter

SeeAlso: AX=4E01h,AX=4E02h,AX=4F00h

Index: installation check;VESA XGA

Format of XGA information buffer:

Offset Size Description (Table 00069)

00h 4 BYTES signature ("VESA")

04h WORD VESA version number

06h DWORD pointer to ASCIZ OEM string

0Ah DWORD environment flags (see #00070)

0Eh WORD number of XGA adapters installed (other VESA XGA functions
require a handle indicating which adapter to use)

10h 240 BYTES reserved

Bitfields for XGA environment flags:

Bit(s) Description (Table 00070)

0-1 system bus (00 = MCA, 01 = ISA, 10 = EISA)

2 bus mastering available

3-31 reserved

-----V-104E01-----

INT 10 - VESA XGA BIOS Extensions - RETURN XGA SUBSYSTEM INFORMATION

AX = 4E01h

DX = XGA handle (0 to number of XGAs-1)

ES:DI -> 256-byte buffer for subsystem information (see #00071)

Return: AL = 4Eh if function supported

AH = status

00h function successful

else error code

SeeAlso: AX=4E00h,AX=4E02h

Format of XGA subsystem information:

Offset Size Description (Table 00071)

00h DWORD pointer to null-terminated board OEM string

04h DWORD capabilities (see #00072)

08h DWORD pointer to 8KB XGA ROM (or NULL)

0Ch DWORD pointer to the XGA memory mapped registers

10h WORD base address of XGA I/O registers (21x0h)

12h DWORD pointer to start of physical video memory

(A000h:0000h or B000h:0000h)

16h DWORD physical address of 4MB aperture (or NULL if none)

1Ah DWORD physical address of 1MB aperture (or NULL if none)

1Eh DWORD physical address of 64KB aperture (or NULL if not enabled)

22h DWORD physical address of OEM aperture (or NULL if none)

26h WORD size of OEM aperture in 64KByte units

28h DWORD pointer to list of video modes

The list is a series of WORDs terminated by FFFFh

2Ch WORD number of 64KB blocks on the board

2Eh DWORD XGA manufacturer ID

byte 0 POS data index 1

byte 1 is index 2

byte 2 is 21xAh index 75h

32h 206 BYTES reserved

Bitfields for XGA capabilities:

Bit(s) Description (Table 00072)

0-1 board bus architecture: 0=MCA, 1=ISA, 3=EISA
 2-3 reserved
 4-6 DMA Channel assigned for acquiring bus mastership (only for ISA bus)
 7 DMA Channel Status (ISA only). enabled if set
 8-31 reserved

-----V-104E02-----

INT 10 - VESA XGA BIOS Extensions - RETURN XGA MODE INFORMATION

AX = 4E02h

CX = Video mode

DX = XGA handle

ES:DI -> 256 byte buffer for mode information (see #00073)

Return: AL = 4Eh if function supported

AH = status

00h function successful

else error code

SeeAlso: AX=4E00h,AX=4E01h

Format of XGA mode information:

Offset Size Description (Table 00073)

00h WORD attributes of the mode (see #00074)
 02h WORD bytes per logical scanline
 04h WORD horizontal resolution in pixels
 06h WORD vertical resolution in scanlines
 08h BYTE character Width in pixels
 09h BYTE character Height in pixels
 0Ah BYTE number of planes
 0Bh BYTE bits per pixels
 0Ch BYTE memory model (see #00075)
 0Dh BYTE number of Image Pages
 0Eh BYTE number of Red bits
 0Fh BYTE bit position of Red bit field
 10h BYTE number of Green bits
 11h BYTE bit position of Green bit field
 12h BYTE number of Blue bits
 13h BYTE bit position of Blue bit field
 14h BYTE number of Reserved bits
 15h BYTE bit position of Reserved bit field
 16h 235 BYTES reserved

Bitfields for XGA mode attributes:

Bit(s) Description (Table 00074)

- 0 set if mode is supported
- 1 reserved
- 2 output is supported by the BIOS
- 3 reserved
- 4 if set this is a graphics mode (VGA registers inactive, XGA active),
if clear this is a text mode (VGA registers active, XGA inactive)

(Table 00075)

Values for XGA video memory model:

- 00h Text Mode
- 01h CGA graphics
- 02h Hercules graphics
- 03h 4-planar graphics
- 04h Packed Pixel
- 05h Non-chain 4, 256 color
- 06h Direct Color
- 07h YUV-24

-----V-104E03-----

INT 10 - VESA XGA BIOS Extensions - SET XGA VIDEO MODE

- AX = 4E03h
- BX = video mode
- CX = other command flags
 - bit 0 If clear the feature connector is set to the default state
- DX = XGA handle
- ES:DI -> 256 byte buffer

Return: AL = 4Eh if function supported

- AH = status
 - 00h function successful
 - else error code

SeeAlso: AH=00h,AX=4E04h,AX=4F02h

-----V-104E04-----

INT 10 - VESA XGA BIOS Extensions - RETURN CURRENT VIDEO MODE

- AX = 4E04h
- DX = XGA handle

Return: AL = 4Eh if function supported

- AH = status
 - 00h function successful
 - BX??? = current mode
 - else error code

SeeAlso: AH=0Fh,AX=4E03h,AX=4F03h

-----V-104E05-----

INT 10 - VESA XGA BIOS Extensions - SET FEATURE CONNECTOR STATE

AX = 4E05h

BX = Feature Connector State (see #00076)

DX = XGA handle

Return: AL = 4Eh if function supported

AH = status

00h function successful

else error code

SeeAlso: AX=4E00h,AX=4E06h

Bitfields for XGA Feature Connector State:

Bit(s) Description (Table 00076)

0 Feature Connector is enabled

1 Feature Connector is in Output Mode rather than Input Mode

2-15 reserved (0)

-----V-104E06-----

INT 10 - VESA XGA BIOS Extensions - RETURN FEATURE CONNECTOR STATE

AX = 4E06h

DX = XGA handle

Return: AL = 4Eh if function supported

AH = status

00h function successful

else error code

BX = Feature Connector State (see #00076)

SeeAlso: AX=4E00h,AX=4E05h

-----V-104F00-----

INT 10 - VESA SuperVGA BIOS (VBE) - GET SuperVGA INFORMATION

AX = 4F00h

ES:DI -> buffer for SuperVGA information (see #00077)

Return: AL = 4Fh if function supported

AH = status

00h successful

ES:DI buffer filled

01h failed

---VBE v2.0---

02h function not supported by current hardware configuration

03h function invalid in current video mode

Desc: determine whether VESA BIOS extensions are present and the capabilities supported by the display adapter

SeeAlso: AX=4E00h,AX=4F01h,AX=7F00h"SOLLEX",AX=A00Ch

Index: installation check;VESA SuperVGA

Format of SuperVGA information:

Offset Size Description (Table 00077)

00h 4 BYTES (ret) signature ("VESA")

(call) VESA 2.0 request signature ("VBE2"), required to receive
version 2.0 info

04h WORD VESA version number (one-digit minor version -- 0102h = v1.2)

06h DWORD pointer to OEM name

"761295520" for ATI

0Ah DWORD capabilities flags (see #00078)

0Eh DWORD pointer to list of supported VESA and OEM video modes

(list of words terminated with FFFFh)

12h WORD total amount of video memory in 64K blocks

---VBE v1.x ---

14h 236 BYTES reserved

---VBE v2.0 ---

14h WORD OEM software version (BCD, high byte = major, low byte = minor)

16h DWORD pointer to vendor name

1Ah DWORD pointer to product name

1Eh DWORD pointer to product revision string

22h WORD (if capabilities bit 3 set) VBE/AF version (BCD)

0100h for v1.0P

24h DWORD (if capabilities bit 3 set) pointer to list of supported
accelerated video modes (list of words terminated with FFFFh)

28h 216 BYTES reserved for VBE implementation

100h 256 BYTES OEM scratchpad (for OEM strings, etc.)

Notes: the list of supported video modes is stored in the reserved portion of
the SuperVGA information record by some implementations, and it may
thus be necessary to either copy the mode list or use a different
buffer for all subsequent VESA calls

not all of the video modes in the list of mode numbers may be
supported, e.g. if they require more memory than currently installed
or are not supported by the attached monitor. Check any mode you
intend to use through AX=4F01h first.

the 1.1 VESA document specifies 242 reserved bytes at the end, so the
buffer should be 262 bytes to ensure that it is not overrun; for
v2.0, the buffer should be 512 bytes

the S3 specific video modes will most likely follow the FFFFh
terminator at the end of the standard modes. A search must then

be made to find them, FFFFh will also terminate this second list
 in some cases, only a "stub" VBE may be present, supporting only
 AX=4F00h; this case may be assumed if the list of supported video
 modes is empty (consisting of a single word of FFFFh)

Bitfields for VESA capabilities:

Bit(s) Description (Table 00078)

- 0 DAC can be switched into 8-bit mode
- 1 non-VGA controller
- 2 programmed DAC with blank bit (i.e. only during blanking interval)
- 3 (VBE v3.0) controller supports hardware stereoscopic signalling
- 3 controller supports VBE/AF v1.0P extensions
- 4 (VBE v3.0) if bit 3 set:
 - =0 stereo signalling via external VESA stereo connector
 - =1 stereo signalling via VESA EVC connector
- 4 (VBE/AF v1.0P) must call EnableDirectAccess to access framebuffer
- 5 (VBE/AF v1.0P) controller supports hardware mouse cursor
- 6 (VBE/AF v1.0P) controller supports hardware clipping
- 7 (VBE/AF v1.0P) controller supports transparent BitBLT
- 8-31 reserved (0)

SeeAlso: #00077,AX=4F09h

-----V-104F01-----

INT 10 - VESA SuperVGA BIOS - GET SuperVGA MODE INFORMATION

AX = 4F01h

CX = SuperVGA video mode (see #04082 for bitfields)

ES:DI -> 256-byte buffer for mode information (see #00079)

Return: AL = 4Fh if function supported

AH = status

00h successful

ES:DI buffer filled

01h failed

Desc: determine the attributes of the specified video mode

SeeAlso: AX=4F00h,AX=4F02h

Bitfields for VESA/VBE video mode number:

Bit(s) Description (Table 04082)

- 15 preserve display memory on mode change
- 14 (VBE v2.0+) use linear (flat) frame buffer
- 13 (VBE/AF 1.0P) VBE/AF initializes accelerator hardware
- 12 reserved for VBE/AF
- 11 (VBE v3.0) user user-specified CRTIC refresh rate values

10-9 reserved for future expansion
 8-0 video mode number (0xxh are non-VESA modes, 1xxh are VESA-defined)

Format of VESA SuperVGA mode information:

Offset Size Description (Table 00079)

00h WORD mode attributes (see #00080)
 02h BYTE window attributes, window A (see #00081)
 03h BYTE window attributes, window B (see #00081)
 04h WORD window granularity in KB
 06h WORD window size in KB
 08h WORD start segment of window A (0000h if not supported)
 0Ah WORD start segment of window B (0000h if not supported)
 0Ch DWORD -> FAR window positioning function (equivalent to AX=4F05h)
 10h WORD bytes per scan line

---remainder is optional for VESA modes in v1.0/1.1, needed for OEM modes---

12h WORD width in pixels (graphics) or characters (text)
 14h WORD height in pixels (graphics) or characters (text)
 16h BYTE width of character cell in pixels
 17h BYTE height of character cell in pixels
 18h BYTE number of memory planes
 19h BYTE number of bits per pixel
 1Ah BYTE number of banks
 1Bh BYTE memory model type (see #00082)
 1Ch BYTE size of bank in KB
 1Dh BYTE number of image pages (less one) that will fit in video RAM
 1Eh BYTE reserved (00h for VBE 1.0-2.0, 01h for VBE 3.0)

---VBE v1.2+ ---

1Fh BYTE red mask size
 20h BYTE red field position
 21h BYTE green mask size
 22h BYTE green field size
 23h BYTE blue mask size
 24h BYTE blue field size
 25h BYTE reserved mask size
 26h BYTE reserved mask position
 27h BYTE direct color mode info
 bit 0: color ramp is programmable
 bit 1: bytes in reserved field may be used by application

---VBE v2.0+ ---

28h DWORD physical address of linear video buffer
 2Ch DWORD pointer to start of offscreen memory

30h WORD KB of offscreen memory
---VBE v3.0 ---
32h WORD bytes per scan line in linear modes
34h BYTE number of images (less one) for banked video modes
35h BYTE number of images (less one) for linear video modes
36h BYTE linear modes: size of direct color red mask (in bits)
37h BYTE linear modes: bit position of red mask LSB (e.g. shift count)
38h BYTE linear modes: size of direct color green mask (in bits)
39h BYTE linear modes: bit position of green mask LSB (e.g. shift count)
3Ah BYTE linear modes: size of direct color blue mask (in bits)
3Bh BYTE linear modes: bit position of blue mask LSB (e.g. shift count)
3Ch BYTE linear modes: size of direct color reserved mask (in bits)
3Dh BYTE linear modes: bit position of reserved mask LSB
3Eh DWORD maximum pixel clock for graphics video mode, in Hz
42h 190 BYTES reserved (0)

Note: while VBE 1.1 and higher will zero out all unused bytes of the buffer,
v1.0 did not, so applications that want to be backward compatible
should clear the buffer before calling

Bitfields for VESA SuperVGA mode attributes:

Bit(s) Description (Table 00080)

0 mode supported by present hardware configuration
1 optional information available (must be =1 for VBE v1.2+)
2 BIOS output supported
3 set if color, clear if monochrome
4 set if graphics mode, clear if text mode

---VBE v2.0+ ---

5 mode is not VGA-compatible
6 bank-switched mode not supported
7 linear framebuffer mode supported
8 double-scan mode available (e.g. 320x200 and 320x240)

---VBE v3.0 ---

9 interlaced mode available
10 hardware supports triple buffering
11 hardware supports stereoscopic display
12 dual display start address support
13-15 reserved

---VBE/AF v1.0P---

9 application must call EnableDirectAccess before calling bank-switching
functions

SeeAlso: #00079

Bitfields for VESA SuperVGA window attributes:

Bit(s) Description (Table 00081)

0 exists
1 readable
2 writable
3-7 reserved

SeeAlso: #00079

(Table 00082)

Values for VESA SuperVGA memory model type:

00h text
01h CGA graphics
02h HGC graphics
03h 16-color (EGA) graphics
04h packed pixel graphics
05h "sequ 256" (non-chain 4) graphics
06h direct color (HiColor, 24-bit color)
07h YUV (luminance-chrominance, also called YIQ)
08h-0Fh reserved for VESA
10h-FFh OEM memory models

SeeAlso: #00079

-----V-104F02-----

INT 10 - VESA SuperVGA BIOS - SET SuperVGA VIDEO MODE

AX = 4F02h

BX = new video mode (see #04082, #00083, #00084)

ES:DI -> (VBE 3.0+) CRTC information block, bit mode bit 11 set
(see #04083)

Return: AL = 4Fh if function supported

AH = status

00h successful
01h failed

Notes: bit 13 may only be set if the video mode is present in the list of
accelerated video modes returned by AX=4F00h

if the DAC supports both 8 bits per primary color and 6 bits, it will
be reset to 6 bits after a mode set; use AX=4F08h to restore 8 bits

SeeAlso: AX=4E03h, AX=4F00h, AX=4F01h, AX=4F03h, AX=4F08h

(Table 00083)

Values for VESA video mode:

00h-FFh OEM video modes (see #00010 at AH=00h)


```
100h 640x400x256
101h 640x480x256
102h 800x600x16
103h 800x600x256
104h 1024x768x16
105h 1024x768x256
106h 1280x1024x16
107h 1280x1024x256
108h 80x60 text
109h 132x25 text
10Ah 132x43 text
10Bh 132x50 text
10Ch 132x60 text
---VBE v1.2+ ---
10Dh 320x200x32K
10Eh 320x200x64K
10Fh 320x200x16M
110h 640x480x32K
111h 640x480x64K
112h 640x480x16M
113h 800x600x32K
114h 800x600x64K
115h 800x600x16M
116h 1024x768x32K
117h 1024x768x64K
118h 1024x768x16M
119h 1280x1024x32K (1:5:5:5)
11Ah 1280x1024x64K (5:6:5)
11Bh 1280x1024x16M
```

```
---VBE 2.0+ ---
120h 1600x1200x256
121h 1600x1200x32K
122h 1600x1200x64K
```

81FFh special full-memory access mode

Notes: the special mode 81FFh preserves the contents of the video memory and gives access to all of the memory; VESA recommends that the special mode be a packed-pixel mode. For VBE 2.0+, it is required that the VBE implement the mode, but not place it in the list of available modes (mode information for this mode can be queried directly, however).

as of VBE 2.0, VESA will no longer define video mode numbers

SeeAlso: #00010,#00011,#00084,#00191

Index: video modes;VESA

(Table 00084)

Values for S3 OEM video mode:

201h 640x480x256
202h 800x600x16
203h 800x600x256
204h 1024x768x16
205h 1024x768x256
206h 1280x960x16
207h 1152x864x256 (Diamond Stealth 64)
208h 1280x1024x16
209h 1152x864x32K
20Ah 1152x864x64K (Diamond Stealth 64)
20Bh 1152x864x4G
211h 640x480x64K (Diamond Stealth 24)
211h 640x400x4G (Diamond Stealth64 Video / Stealth64 Graphics)
212h 640x480x16M (Diamond Stealth 24)
301h 640x480x32K

Note: these modes are only available on video cards using S3's VESA driver

SeeAlso: #00083,#00191,#00732 at INT 1A/AX=B102h

Index: video modes;S3

Format of VESA VBE CRTIC Information Block:

Offset Size Description (Table 04083)

00h WORD total number of pixels horizontally
02h WORD horizontal sync start (in pixels)
04h WORD horizontal sync end (in pixels)
06h WORD total number of scan lines
08h WORD vertical sync start (in scan lines)
0Ah WORD vertical sync end (in scan lines)
0Ch BYTE flags (see #04084)
0Dh DWORD pixel clock, in Hz
11h WORD refresh rate, in 0.01 Hz units
 this field MUST be set to $\text{pixel_clock} / (\text{HTotal} * \text{VTotal})$,
 even though it may not actually be used by the VBE
 implementation
13h 40 BYTES reserved

Bitfields for VESA VBE CRTIC Information Block flags:

Bit(s) Description (Table 04084)

- 0 enable double scanning
- 1 enable interlacing
- 2 horizontal sync polarity (0 positive, 1 negative)
- 3 vertical sync polarity (0 positive, 1 negative)

SeeAlso: #04083

-----V-104F03-----

INT 10 - VESA SuperVGA BIOS - GET CURRENT VIDEO MODE

AX = 4F03h

Return: AL = 4Fh if function supported

AH = status

00h successful

BX = video mode (see #00083,#00084)

bit 13: VBE/AF v1.0P accelerated video mode

bit 14: linear frame buffer enabled (VBE v2.0+)

bit 15: don't clear video memory

01h failed

SeeAlso: AH=0Fh,AX=4E04h,AX=4F02h

-----V-104F04-----

INT 10 - VESA SuperVGA BIOS - SAVE/RESTORE SuperVGA VIDEO STATE

AX = 4F04h

DL = subfunction

00h get state buffer size

Return: BX = number of 64-byte blocks needed

01h save video states

ES:BX -> buffer

02h restore video states

ES:BX -> buffer

CX = states to save/restore (see #00085)

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

SeeAlso: AH=1Ch,AX=5F90h,AX=5FA0h

Bitfields for VESA SuperVGA states to save/restore:

Bit(s) Description (Table 00085)

- 0 video hardware state
- 1 video BIOS data state
- 2 video DAC state
- 3 SuperVGA register state

SeeAlso: #00048,#00186

-----V-104F05-----

INT 10 - VESA SuperVGA BIOS - CPU VIDEO MEMORY CONTROL

AX = 4F05h

BH = subfunction

00h select video memory window

DX = window address in video memory (in granularity units)

01h get video memory window

Return: DX = window address in video memory (in gran. units)

BL = window number

00h window A

01h window B

ES = selector for memory-mapped registers (VBE 2.0+, when called from
32-bit protected mode)

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

Note: when using an accelerated video mode under VBE/AF v1.0P, the
application must call EnableDirectAccess before switching banks if
bit 9 of the video mode attributes flag is set (see #00080)

BUG: Phoenix S3 Trio64V+ v1.02-02 reportedly returns garbage in DX for
subfunction 01h

SeeAlso: AX=4F01h,AX=4F06h,AX=4F07h,AX=7000h/BX=0004h

-----V-104F06-----

INT 10 - VESA SuperVGA BIOS v1.1+ - GET/SET LOGICAL SCAN LINE LENGTH (PIXELS)

AX = 4F06h

BL = subfunction

00h set scan line length

CX = desired width in pixels

01h get scan line length

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

BX = bytes per scan line

CX = number of pixels per scan line

DX = maximum number of scan lines

Notes: if the desired width is not achievable, the next larger width will be
set

the scan line may be wider than the visible area of the screen

this function is valid in text modes, provided that pixel values are multiplied by the character cell width/height

SeeAlso: AX=4F01h,AX=4F05h,AX=4F06h/BL=02h,AX=4F06h/BL=03h,AX=4F07h

-----V-104F06BL02-----

INT 10 - VESA SuperVGA BIOS v2.0+ - SET LOGICAL SCAN LINE LENGTH (BYTES)

AX = 4F06h

BL = 02h

CX = desired scanline width in bytes

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

BX = bytes per scan line

CX = number of pixels per scan line

DX = maximum number of scan lines

Notes: if the desired width is not achievable, the next larger width will be set

the scan line may be wider than the visible area of the screen

this function is valid in text modes, provided that pixel values are multiplied by the character cell width/height

SeeAlso: AX=4F01h,AX=4F05h,AX=4F06h,AX=4F06h/BL=03h,AX=4F07h

-----V-104F06BL03-----

INT 10 - VESA SuperVGA BIOS v2.0+ - GET MAXIMUM SCAN LINE LENGTH

AX = 4F06h

BL = 03h

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

BX = maximum bytes per scan line

CX = maximum number of pixels per scan line

Notes: this function is valid in text modes, provided that pixel values are multiplied by the character cell width/height

SeeAlso: AX=4F01h,AX=4F05h,AX=4F06h,AX=4F06h/BL=02h,AX=4F07h

-----V-104F07-----

INT 10 - VESA SuperVGA BIOS v1.1+ - GET/SET DISPLAY START

AX = 4F07h

BL = subfunction

00h,80h set display start

CX = leftmost displayed pixel in scan line

DX = first displayed scan line

```
    01h get display start
Return: BH = 00h
    CX = leftmost displayed pixel in scan line
    DX = first displayed scan line
    BH = 00h (reserved)
Return: AL = 4Fh if function supported
    AH = status
```

```
    00h successful
    01h failed
```

Notes: this function is valid in text modes, provided that values are multiplied by the character cell width/height
subfunction 80h waits until vertical retrace before changing the display start address (VBE v2.0+)

SeeAlso: AX=4F01h,AX=4F05h,AX=4F06h

-----V-104F07-----

INT 10 - VESA SuperVGA BIOS v3.0+ - SCHEDULED DISPLAY START

```
    AX = 4F07h
```

```
    BL = subfunction
```

```
    02h,82h (VBE v3.0) schedule display start at next vertical retrace
```

```
    ECX = display start address in bytes
```

```
    03h,83h schedule stereoscopic display start
```

```
    ECX = left image start address (in bytes)
```

```
    EDX = right image start address (in bytes)
```

```
    04h get scheduled display start status
```

```
Return: CX = start status
```

```
    (00h no display flip, else display has flipped)
```

```
Return: AL = 4Fh if function supported
```

```
    AH = status
```

```
    00h successful
```

```
    01h failed
```

SeeAlso: AX=4F07h"DISPLAY START",AX=4F07h"STEREOSCOPIC"

-----V-104F07-----

INT 10 - VESA SuperVGA BIOS v3.0+ - ENABLE/DISABLE STEREOSCOPIC MODE

```
    AX = 4F07h
```

```
    BL = subfunction
```

```
    05h enable stereoscopic mode
```

```
    06h disable stereoscopic mode
```

```
Return: AL = 4Fh if function supported
```

```
    AH = status
```

```
    00h successful
```

```
    01h failed
```

Note: when stereoscopic mode is enabled, the display start address will automatically alternate between the left and right images on every vertical retrace

SeeAlso: AX=4F07h"DISPLAY START",AX=4F07h"SCHEDULED"

-----V-104F08-----

INT 10 - VESA SuperVGA BIOS v1.2+ - GET/SET DAC PALETTE CONTROL

AX = 4F08h

BL = subfunction

00h set DAC palette width

BH = desired number of bits per primary color

01h get DAC palette width

Return: AL = 4Fh if function supported

AH = status

00h successful

BH = current number of bits per primary (06h = standard VGA)

01h failed

-----V-104F09-----

INT 10 - VESA SuperVGA BIOS v2.0+ - GET/SET PALETTE ENTRIES

AX = 4F09h

BL = subfunction

00h set (primary) palette

01h get (primary) palette

02h set secondary palette data

03h get secondary palette data

80h set palette during vertical retrace

CX = number of entries to change

DX = starting palette index

ES:DI -> palette buffer, array of DAC entries (see #00086)

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

SeeAlso: AX=4F0Ah,AX=D000h"S3VBE"

Format of VESA VBE palette entry:

Offset Size Description (Table 00086)

00h BYTE red

01h BYTE green

02h BYTE blue

03h BYTE alpha or alignment byte

-----V-104F0ABL00-----

INT 10 - VESA SuperVGA BIOS v2.0+ - GET PROTECTED-MODE INTERFACE

AX = 4F0Ah

BL = 00h

Return: AL = 4Fh if function supported

AH = status

00h successful

ES:DI -> protected-mode table (see #00087)

CX = length of table in bytes, included protected-mode code

01h failed

SeeAlso: AX=4F00h,AX=4F05h,AX=4F07h,AX=4F09h

Format of VESA VBE 2.0 protected-mode table:

Offset Size Description (Table 00087)

00h WORD offset (within table) of protected-mode code for Function 5

(Set Window)

02h WORD offset of protected-mode code for Function 7 (Set Disp Start)

04h WORD offset of protected-mode code for Function 9 (Set Primary
Palette)

06h WORD offset (within table) of list of I/O ports and memory locations
for which I/O privilege may be required in protected mode
(0000h if no list) (see #04085)

08h var code and optional port/memory list

Format of VESA VBE 2.0+ protected-mode port/memory list:

Offset Size Description (Table 04085)

00h WORD I/O port number

...

WORD FFFFh marking end of I/O port list

DWORD memory region base address

WORD memory region limit (one less than length)

WORD FFFFh marking end of memory region list

Note: the memory region is optional; if not present, there will be a word
of FFFFh immediately following the FFFFh marking the end of the I/O
port list

SeeAlso: #00087

-----V-104F0A-----

INT 10 0 - VESA SuperVGA BIOS v2.0 beta - GET PROTECTED-MODE CODE

AX = 4F0Ah

BX = function to be copied (see #00088)

Return: AL = 4Fh if function supported

AH = status

00h successful
 CX = number of bytes to copy
 ES:DI -> code for requested function
 01h failed

SeeAlso: AX=4F09h,AX=D000h"S3VBE"

(Table 00088)

Values for VESA VBE protected-mode function identifiers:

0001h set bank (see AX=4F05h)
 0002h set display start (see AX=4F07h)
 0003h set palette (see AX=4F09h)

-----V-104F0BBL00-----

INT 10 - VESA SuperVGA BIOS v3.0 - GET NEAREST PIXEL CLOCK

AX = 4F0Bh
 BL = 00h get closest pixel clock
 ECX = requested clock frequency in Hz
 DX = video mode number

Return: AL = 4Fh if function supported

AH = status
 00h successful
 ECX = nearest clock rate
 01h failed

-----V-104F0B-----

INT 10 - VESA SuperVGA BIOS - GET VBE/AF v1.0P DEVICE CONTEXT BUFFER

AX = 4F0Bh
 BL = subfunction
 00h get length of device context buffer
 01h initialize device context buffer
 ES:DI -> real-mode address of buffer (see #00089)

Return: AL = 4Fh if function supported

AH = status
 00h successful
 CX = number of bytes required for buffer (if BL=00h on entry)
 ES:DI buffer initialized (if BL=01h on entry)
 01h failed

Note: this interface description is derived from the draft VBE/AF proposal
 (version 1.0P, document revsion 0.12P, dated 13jan95)

SeeAlso: AX=4F00h,AX=4F01h,AX=4F17h

Format of VBE/AF v1.0P Device Context buffer:

Offset Size Description (Table 00089)

00h WORD maximum X coordinate in frame buffer (0 = left edge)
Note: application must never pass X coordinate greater than
this value to the SetClipRect function (values are
NOT range-checked!)

02h WORD maximum Y coordinate in frame buffer (0 = top edge)
Note: application must never pass Y coordinate greater than
this value to the SetClipRect function

04h DWORD -> "SetForeColor" to set foreground color/mix (see #00091)

08h DWORD -> "SetBackColor" to set background color/mix (see #00092)

0Ch DWORD -> "SetClipRect" set hardware clipping rectangle (see #00093)

10h DWORD -> function "DrawScan" to draw a single scan line

14h DWORD -> function "DrawScanList" to draw a list of scan lines

18h DWORD -> function "DrawRect" to draw a solid rectangle

1Ch DWORD -> function "DrawLine" to draw a solid line

20h DWORD -> function "DrawPattScan" to draw a patterned scan line

24h DWORD -> function "BitBlt" to perform screen-to-screen BitBLT

28h DWORD -> "TransBitBlt" to perform transparent screen-to-screen BitBLT

2Ch DWORD -> "MonoBitBlt" to monochrome expansion screen-to-screen BitBLT

30h DWORD -> function "SetCursor" to download hardware cursor image

34h DWORD -> function "SetCursorPos" to set hardware cursor position

38h DWORD -> function "SetCursorColor" to set hardware cursor color

3Ch DWORD -> function "ShowCursor" to show/hide hardware cursor

40h DWORD -> function "WaitTillIdle" to wait until graphics engine idle

44h DWORD -> "EnableDirectAccess" to enable direct framebuffer access

48h DWORD -> "DisableDirectAccess" to disable direct framebuffer access

4Ch DWORD -> "BankSwitchCB" bank-switching callback function
(set by application, may simply point at 32-bit VBE 2.0 bank
switching entry point)

50h WORD "VidMemSel" selector for video memory (must be set by app)

52h WORD "IOPortsOff" offset of I/O ports table in context buffer

54h WORD "IOMemoryOff" offset of I/O memory table in context buffer

56h WORD "IOMemSel1" selector for first I/O memory area specified by
I/O memory table (must be set by application)

58h WORD "IOMemSel2" selector for second I/O memory area

5Ah WORD "IOMemSel3" selector for third I/O memory area

5Ch WORD "IOMemSel4" selector for fourth I/O memory area

5Eh N BYTES device-specific state buffer
N WORDs I/O port access table (list of ports, last entry is FFFFh)
N DWORDs I/O memory access table (list of physical-address/length pairs,
last entry is FFFFFFFFh)
N BYTES 32-bit code for VBE/AF v1.0P function

Notes: all function pointers are offsets into the 32-bit code within the context buffer, and should be updated to point at the actual functions after the application has copied the buffer from the real-mode memory used for the interrupt call into its own flat-model memory space
any functions which are not supported by the hardware have function pointers which are set to 00000000h initially

(Table 00090)

Values for VBE/AF v1.0P color mixing mode:

00h replace
01h XOR with existing pixel
02h OR with existing pixel
03h AND with existing pixel

SeeAlso: #00091,#00090

(Table 00091)

Call VBE/AF v1.0P function "SetForeColor" with:

ES:EDI -> device context buffer (see #00089)
EAX = new foreground color (format varies by video mode)
BL = new foreground mixing mode (see #00090)

SeeAlso: #00089,#00092,#00107

(Table 00092)

Call VBE/AF v1.0P function "SetbackColor" with:

ES:EDI -> device context buffer (see #00089)
EAX = new background color (format varies by video mode)
BL = new background mixing mode (see #00090)

SeeAlso: #00089,#00091,#00093

(Table 00093)

Call VBE/AF v1.0P function "SetClipRect" with:

ES:EDI -> device context buffer (see #00089)
EAX = left clipping coordinate
EBX = top clipping coordinate
ECX = right clipping coordinate
EDX = bottom clipping coordinate

SeeAlso: #00089,#00092,#00094,#00107

(Table 00094)

Call VBE/AF v1.0P function "DrawScan" with:

ES:EDI -> device context buffer (see #00089)

EAX = X1 coordinate

EBX = Y coordinate

ECX = X2 coordinate

Desc: draw a solid horizontal line from (X1,Y) to (X2,Y) in the currently-active foreground color and mix, omitting the pixel at the largest X coordinate

SeeAlso: #00089,#00093,#00095,#00098,#00107

(Table 00095)

Call VBE/AF v1.0P function "DrawScanList" with:

ES:EDI -> device context buffer (see #00089)

EAX = Y coordinate of first line

EBX = length of scan list

DS:ESI -> scanline list X coordinates (2N WORDs)

Desc: draw multiple solid horizontal lines at successive Y coordinates, using the currently-active foreground color and mix, omitting the pixel at the largest X coordinate for each line

Note: the scanline list consists of pairs of X coordinates; because the last pixel is omitted, a scanline will be skipped if X1==X2

SeeAlso: #00089,#00094,#00096,#00107

(Table 00096)

Call VBE/AF v1.0P function "DrawRect" with:

ES:EDI -> device context buffer (see #00089)

EAX = left coordinate

EBX = top coordinate

ECX = right coordinate

EDX = bottom coordinate

Desc: draw a solid rectangle in the currently-active foreground color and mix, omitting the rightmost X coordinate and bottom-most scan line

Note: results are undefined if EAX>ECX or EBX>EDX; nothing will be drawn if EAX=ECX or EBX=EDX

SeeAlso: #00089,#00094,#00095,#00097

(Table 00097)

Call VBE/AF v1.0P function "DrawLine" with:

ES:EDI -> device context buffer (see #00089)

EAX = ??? (specified this way in the draft documentation!)

EBX = ???

ECX = ???

EDX = ???

ESI = ???

Desc: draw a solid line in the currently-active foreground color and mix,
given the Bresenham parameters

SeeAlso: #00089,#00094,#00096,#00098

(Table 00098)

Call VBE/AF v1.0P function "DrawPattScan" with:

ES:EDI -> device context buffer (see #00089)

EAX = X1 coordinate

EBX = Y coordinate

ECX = X2 coordinate

DL = 8-bit stipple pattern

Desc: draw a patterned horizontal line from (X1,Y) to (X2,Y) in the
currently-active foreground color and mix, omitting the pixel at
the largest X coordinate

Note: for each pixel in the line, if DL bit (X mod 8) is set, the pixel is
drawn, and left untouched if the bit is clear

SeeAlso: #00089,#00096,#00097,#00094,#00099,#00107

(Table 00099)

Call VBE/AF v1.0P function "BitBlt" with:

ES:EDI -> device context buffer (see #00089)

DS:ESI -> BitBlt parameter block (see #00100)

BL = mix operation

00h replace

01h XOR

02h OR

03h AND

SeeAlso: #00089,#00094,#00098,#00101,#00107

Format of VBE/AF BitBlt parameter block:

Offset Size Description (Table 00100)

00h WORD left coordinate of source rectangle

02h WORD top coordinate of source rectangle

04h WORD right coordinate of source rectangle

06h WORD bottom coordinate of source rectangle

08h WORD left coordinate of destination rectangle

0Ah WORD top coordinate of destination rectangle

0Ch BYTE horizontal direction: 00h = decrement X, 01h = increment X

0Dh BYTE vertical direction: 00h = decrement Y, 01h = increment Y

Notes: the rightmost pixel(s) and bottom-most scan line are not copied
the horizontal/vertical direction flags are used to ensure correct
copies when the source and destination rectangles overlap

SeeAlso: #00099

(Table 00101)

Call VBE/AF v1.0P function "TransBitBlt" with:

ES:EDI -> device context buffer (see #00089)
DS:ESI -> BitBlt parameter block (see #00100)

BL = mix operation

00h replace
01h XOR
02h OR
03h AND

ECX = transparent color

Desc: copy a rectangular area from one location to another, treating pixels
with the specified color as transparent (leaving the destination
unchanged)

SeeAlso: #00089,#00099,#00102,#00107

(Table 00102)

Call VBE/AF v1.0P function "MonoBitBlt" with:

ES:EDI -> device context buffer (see #00089)
DS:ESI -> BitBlt parameter block (see #00100)

BL = mix operation

00h replace
01h XOR
02h OR
03h AND

BH = bit-plane from which to read

Desc: copy a rectangular area from one location to another, expanding a
single bit of each source pixel

SeeAlso: #00089,#00099,#00101

(Table 00103)

Call VBE/AF v1.0P function "SetCursor" with:

ES:EDI -> device context buffer (see #00089)
DS -> application's data segment
ESI -> 32x32 cursor data, in Windows 3.1 cursor-file format
(32 DWORDs of XOR mask followed by 32 DWORDs of AND mask)

SeeAlso: #00089,#00094,#00104,#00105

(Table 00104)

Call VBE/AF v1.0P function "SetCursorPos" with:

ES:EDI -> device context buffer (see #00089)

EAX = cursor X coordinate

EBX = cursor Y coordinate

CL = cursor's X hotspot

CH = cursor's Y hotspot

SeeAlso: #00089,#00103,#00105

(Table 00105)

Call VBE/AF v1.0P function "SetCursorColor" with:

ES:EDI -> device context buffer (see #00089)

AL = 8-bit color index, or color's Red value

AH = color's Green value (16-bpp, 24-bpp, 32-bpp modes)

BL = color's Blue value (16-bpp, 24-bpp, 32-bpp modes)

SeeAlso: #00089,#00103,#00104,#00106

(Table 00106)

Call VBE/AF v1.0P function "ShowCursor" with:

ES:EDI -> device context buffer (see #00089)

AL = new visibility (00h hide cursor, 01h show cursor)

SeeAlso: #00089,#00103,#00104,#00105

(Table 00107)

Call VBE/AF v1.0P function "WaitTillIdle" with:

ES:EDI -> device context buffer (see #00089)

Desc: pause until the hardware accelerator has completed all pending operations

SeeAlso: #00089,#00108,#00109

(Table 00108)

Call VBE/AF v1.0P function "EnableDirectAccess" with:

ES:EDI -> device context buffer (see #00089)

SeeAlso: #00089,#00109,#00107,AX=4F05h,#00078,#00080

(Table 00109)

Call VBE/AF v1.0P function "DisableDirectAccess" with:

ES:EDI -> device context buffer (see #00089)

SeeAlso: #00089,#00108

-----V-104F0C-----

INT 10 - VESA SuperVGA BIOS - RESERVED FOR FUTURE SVGA FUNCTIONS

AX = 4F0Ch-4F0Fh

SeeAlso: AX=4F00h

-----V-104F0B-----

INT 10 - Diamond Viper V330 - ???

AX = 4F0Bh

ECX = ???

???

Return: AL = 4Fh if function supported

AH = status

00h successful

ECX = ???

???

01h failed

Note: in BIOS v1.62, ECX is in multiples of 10000 (decimal) for both input and output, and this function never fails

-----p-104F10BL00-----

INT 10 - VESA VBE/PM (Power Management) v1.0+ - GET CAPABILITIES

AX = 4F10h

BL = 00h

ES:DI = 0000h:0000h (reserved for future use as pointer to info block)

Return: AL = 4Fh if function supported (installed)

AH = call status

00h successful

else failed

---if successful---

BL = VBE/PM version (bits 7-4: major, bits 3-0: minor)

BH = supported states (see #00110)

ES:DI unchanged

Index: installation check;VESA VBE/Power Management|VESA DPMS

SeeAlso: AX=4F10h/BL=01h,AX=4F10h/BL=02h

Bitfields for VESA VBE/PM supported power states:

Bit(s) Description (Table 00110)

0 standby

1 suspend

2 off

3 reduced on

-----p-104F10BL01-----

INT 10 - VESA VBE/PM (Power Management) v1.0+ - SET DISPLAY POWER STATE

AX = 4F10h

BL = 01h
BH = new state (see #00111)
Return: AL = 4Fh if function supported
AH = call status
 00h successful
 else failed
SeeAlso: AX=4F10h/BL=00h,AX=4F10h/BL=02h,AX=A00Ch

(Table 00111)

Values for VESA VBE/PM power state:

00h On
01h standby
02h suspend
04h Off
08h reduced On (for flat screens)

-----p-104F10BL02-----

INT 10 - VESA VBE/PM (Power Management) v1.0+ - GET DISPLAY POWER STATE

AX = 4F10h
BL = 02h

Return: AL = 4Fh if function supported

AH = call status
 00h successful
 BH = current power state (see #00111)
 else failed

SeeAlso: AX=4F10h/BL=00h,AX=4F10h/BL=01h,AX=A00Dh

-----V-104F11-----

INT 10 - VESA VBE/FP (Flat Panel Interface) - API

AX = 4F11h
BL = function
 00h installation check / get capabilities
 other registers vary by function

Return: varies by function

-----V-104F12-----

INT 10 - VESA VBE/CI (Cursor Interface) - API

AX = 4F12h
BL = function
 other registers vary by function

Return: varies by function

SeeAlso: AX=4F4Dh

-----s-104F13BX0000-----

INT 10 - VESA VBE/AI (Audio Interface) - INSTALLATION CHECK

```
AX = 4F13h
BX = 0000h
Return: AL = 4Fh if function supported
      AH = status
      00h successful
      BX = version (0010h for 1.00 -- note nonstandard value!)
      01h failed
```

SeeAlso: AX=4F13h/BX=0001h,AX=4F13h/BX=0005h

-----s-104F13BX0001-----

INT 10 - VESA VBE/AI (Audio Interface) - LOCATE DEVICE

```
AX = 4F13h
BX = 0001h
CX = 0000h ???
DX = type of device
      0001h Wave device
      0002h MIDI device
      0003h Volume device
```

```
Return: AL = 4Fh if function supported
      AH = status
      00h successful
      CX = handle [returned 1 for DX=1 and 3, and 2 for DX=2]
      01h failed
```

SeeAlso: AX=4F13h/BX=0000h,AX=4F13h/BX=0002h

-----s-104F13BX0002-----

INT 10 - VESA VBE/AI (Audio Interface) - QUERY DEVICE

```
AX = 4F13h
BX = 0002h
CX = handle
DX = query
      0001h return length of GeneralDeviceClass
      0002h return copy of GeneralDeviceClass (see #00112)
      0003h return length of Volume Info Structure
      0004h return copy of Volume Info Structure (see #00122)
      0005h return length of Volume Services Structure
      0006h return copy of Volume Services Structure (see #00124)
      0007h-000Fh reserved
      0010h-FFFFh device-specific
```

SI:DI -> buffer (functions 0002h,0004h,0006h)

```
Return: AL = 4Fh if function supported
      AH = status
      00h successful
```

SI:DI = length (functions 1,3,5)
 SI:DI buffer filled (functions 2,4,6)

01h failed

Note: functions 0003h to 0006h are only supported for the Volume device

Format of GeneralDeviceClass structure:

Offset	Size	Description (Table 00112)
00h	4 BYTES	name of the structure ("GENI")
04h	DWORD	structure length
08h	WORD	type of device (1=Wave, 2=MIDI)
0Ah	WORD	version of VESA driver support (0100h for 1.00)
10h	var	for CX=handle for Wave device: Wave Info structure (see #00113) some bytes ???
		for CX=handle for MIDI device: MIDI Info Structure (see #00118) first 8 bytes of MIDI Service Structure ???

SeeAlso: #00122,#00124

Format of WAVE Info Structure:

Offset	Size	Description (Table 00113)
00h	4 BYTES	name of the structure ("WAVI")
04h	DWORD	structure length [0000007Eh]
08h	DWORD	driver software version [00000003h]
0Ch	32 BYTES	vendor name, etc. (ASCIZ string)
2Ch	32 BYTES	vendor product name
4Ch	32 BYTES	vendor chip/hardware description
6Ch	BYTE	installed board number
6Dh	3 BYTES	unused data
70h	DWORD	feature bits (see #00114)
74h	WORD	user determined preference field
76h	WORD	memory required for driver use [0200h]
78h	WORD	number of timer tick callbacks per second [0000h]
7Ah	WORD	channels: 1 = mono, 2 = stereo stereo is assumed to be interleaved data
7Ch	WORD	bitfield of max sample sizes (see #00115)

SeeAlso: #00118

Bitfields for Wave feature bits:

Bit(s)	Description (Table 00114)
0	8000hz Mono Playback

```
1 8000hz Mono Record
2 8000hz Stereo Record
3 8000hz Stereo Playback
4 8000hz Full Duplex Play/Record
5 11025hz Mono Playback
6 11025hz Mono Record
7 11025hz Stereo Record
8 11025hz Stereo Playback
9 11025hz Full Duplex Play/Record
10 22050hz Mono Playback
11 22050hz Mono Record
12 22050hz Stereo Record
13 22050hz Stereo Playback
14 22050hz Full Duplex Play/Record
15 44100hz Mono Playback
16 44100hz Mono Record
17 44100hz Stereo Record
18 44100hz Stereo Playback
19 44100hz Full Duplex Play/Record
20-26 reserved (0)
27 driver must pre-handle the data
28 Variable Sample mono playback
29 Variable Sample stereo playback
30 Variable Sample mono record
31 Variable Sample stereo record
```

(Table 00115)

Values for Sample data size:

```
01h 8bit play
02h 16bit play
10h 8bit record
20h 16bit record
```

Format of WAVE Audio Services structure:

```
Offset Size Description (Table 00116)
00h 4 BYTES name of the structure
04h DWORD structure length
08h 16 BYTES for future expansion
---entry points (details???)---
18h DWORD DeviceCheck
    11h compression (see also #00117)
```

12h driver state
13h get current pos
14h sample rate
15h set preference
16h get DMA,IRQ
17h get IO address
18h get mem address
19h get mem free
1Ah full duplex
1Bh get block size
1Ch get PCM format
1Dh enable PCM format
80h-.. vendors can add DevChks above 0x80

1Ch DWORD PCMInfo
20h DWORD PlayBlock
24h DWORD PlayCont
28h DWORD RecordBlock
2Ch DWORD RecordCont
30h DWORD PauseIO
34h DWORD ResumeIO
38h DWORD StopIO
3Ch DWORD WavePrepare
40h DWORD WaveRegister
44h DWORD GetLastError
01h unsupported feature/function
02h bad sample rate
03h bad block length
04h bad block address
05h app. missed an IRQ
06h don't understand the PCM size/format
80h-.. vendors specific errors
48h DWORD TimerTick
4Ch DWORD ApplPSyncCB: Callback: play filled in by the app
50h DWORD ApplRSyncCB: Callback: rec filled in by the app

SeeAlso: #00120,#00124

(Table 00117)

Values for type of compression:

01h IMA play
02h ALAW play
03h ULAW play

11h IMA record
12h ALAW record
13h ULAW record

Format of MIDI Info Structure:

Offset Size Description (Table 00118)

00h 4 BYTES name of the structure ("MIDI")
04h DWORD structure length
08h DWORD driver software version [00000003h]
0Ch 32 BYTES vendor name, etc. (ASCIZ string)
2Ch 32 BYTES vendor product name
4Ch 32 BYTES vendor chip/hardware description
6Ch BYTE installed board number
6Dh 3 BYTES unused data
70h 14 BYTES the patch library file name [OPL2.BNK 00..]
7Eh DWORD feature bits (see #00119)
80h WORD user determined preference field
82h WORD memory required for driver use
84h WORD # of timer tick callbacks per second
86h WORD max # of tones (voices, partials)

SeeAlso: #00112,#00120,#00122

Bitfields for MIDI feature bits:

Bit(s) Description (Table 00119)

0-3 reserved for GM extensions
4 Transmitter/Receiver only
5 Patches preloaded
6 MIDI receive has time stamp
8 MIDI interrupt driven input supported
9 MIDI polled input supported
10 MIDI remote patches supported

Format of MIDI Service structure:

Offset Size Description (Table 00120)

00h 4 BYTES name of the structure ("MIDS")
04h DWORD structure length
08h 16 WORDs patches loaded table bit field
28h 16 BYTES for future expansion

---entry points (details???)---

38h DWORD device check
11h return available tones

12h return TRUE/FALSE if patch is understood
13h set preference
14h allow/disallow voice stealing
15h get FIFO sizes
16h get DMA,IRQ
17h get IO address
18h get mem address
19h get mem free
80h-.. vendors can add DevChks above 0x80
3Ch DWORD global reset
40h DWORD MIDI msg
44h DWORD poll MIDI
48h DWORD preload patch
4Ch DWORD unload patch
50h DWORD timer tick
54h DWORD get last error
01h unsupported feature/function
02h unknown patch type (see #00121)
03h all tones are used
04h messages are out of sync
05h an incoming patch was incomplete
06h an incoming patch couldn't be stored
07h had to drop an incoming byte
08h driver is failing a patch download
80h-.. vendors specific errors
58h DWORD Patch Block free callback
5Ch DWORD MIDI byte avail. callback
SeeAlso: #00116,#00124

(Table 00121)

Values for MIDI Registered Patch Types:

10h OPL2
11h OPL3

Format of Volume Info Structure:

Offset	Size	Description (Table 00122)
00h	4 BYTES	name of the structure ("VOLI")
04h	DWORD	structure length (00000092h)
08h	DWORD	driver software version [00000001h]
0Ch	32 BYTES	vendor name, etc. (ASCIZ string)
2Ch	32 BYTES	vendor product name

4Ch 32 BYTES vendor chip/hardware description
6Ch BYTE installed board number (0 for 1st/only board)
6Dh 3 BYTES unused data (0)
70h 24 BYTES text name of the mixer channel
88h DWORD features bits (see #00123)
8Ch WORD minimum volume setting
8Eh WORD maximum volume setting
90h WORD attenuation/gain crossover

SeeAlso: #00112,#00124

Bitfields for Volume feature bits:

Bit(s) Description (Table 00123)

0 Stereo Volume control available
2 Low Pass Filter is available
3 High Pass Filter is available
4 Parametric Tone Control is available
5 selectable output paths
8 Azimuth Field positioning supported
9 Phi Field positioning supported
10-30 unused???
31 Master Volume device

Format of Volume Services Structure:

Offset Size Description (Table 00124)

00h 4 BYTES name of the structure ("VOLS")
04h DWORD structure length (00000038h)
08h 16 BYTES 16 bytes for future expansion (0)

---entry points (details???)---

18h DWORD device check
 0011h filter range
 0012h filter setting
 0013h filter current
 0014h tone range
 0015h tone setting
 0016h tone current
 0017h path
 0018h get IO address
0080h-.. vendors can add DevChks above 0x80
1Ch DWORD set vol to an absolute setting
 01h User master volume setting
 02h application master volume setting

20h DWORD set 3D volume
24h DWORD tone control
28h DWORD filter control
2Ch DWORD output path
30h DWORD reset channel
34h DWORD get last error
 01h unsupported feature/function
 02h out of range parameter value
 80h+ vendor-specific errors

SeeAlso: #00116, #00120

-----s-104F13BX0003-----

INT 10 - VESA VBE/AI (Audio Interface) - OPEN DEVICE

AX = 4F13h
BX = 0003h
CX = handle
DX = API set (16/32-bit)
SI = segment ???

Return: AL = 4Fh if function supported

AH = status
00h successful
 SI: CX -> memory ???
01h failed

SeeAlso: AX=4F13h/BX=0000h, AX=4F13h/BX=0002h, AX=4F13h/BX=0004h

-----s-104F13BX0004-----

INT 10 - VESA VBE/AI (Audio Interface) - CLOSE DEVICE

AX = 4F13h
BX = 0004h
CX = handle

Return: AL = 4Fh if function supported

AH = status
00h successful
01h failed

SeeAlso: AX=4F13h/BX=0000h, AX=4F13h/BX=0003h, AX=4F13h/BX=0005h

-----s-104F13BX0005-----

INT 10 - VESA VBE/AI (Audio Interface) - UNINSTALL DRIVER

AX = 4F13h
BX = 0005h

Return: AL = 4Fh if function supported

AH = status
00h successful
01h failed

SeeAlso: AX=4F13h/BX=0000h,AX=4F13h/BX=0006h

-----s-104F13BX0006-----

INT 10 - VESA VBE/AI (Audio Interface) - DRIVER CHAIN/UNCHAIN

AX = 4F13h

BX = 0006h

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

SeeAlso: AX=4F13h/BX=0000h,AX=4F13h/BX=0005h

-----104F14-----

INT 10 - VESA OEM Extensions - API

AX = 4F14h

BL = function

00h installation check / get capabilities

other registers vary by function

Return: varies by function

Note: this function is provided for OEMs to implement system-specific
functionality in a VESA-compatible manner

-----104F14-----

INT 10 - VESA OEM Extensions - Matrox Millenium (BIOS v2.1)

AX = 4F14h

BL = function

00h installation check / get capabilities

ES:DI -> 256-byte buffer for Matrox-specific information

(see #00125)

Return: ES:DI buffer filled

01h read/write MGA indexed data register

BH = direction (00h = write, else read)

CL = register number (80h for most-recently used register)

CH = new value, if writing

Return: CH = current value, if reading

02h read/write register in MGA control space

BH = direction and size

bit 4: read register instead of writing

bits 1-0: data size (00 = BYTE, 01 = WORD, 10 = DWORD)

SI = register address

CL/CX/ECX = new value if writing

Return: CL/CX/ECX = current value if reading

03h read PINS byte

SI = offset of byte within PINS data (see #00126)

```

Return: CL = value of specified PINS byte
    04h ??? (related to PLL programming)
DX = ???
Return: CL,EDX destroyed
    05h read BIOS byte
SI = offset of byte relative to start of BIOS image
Return: CL = specified byte
    else
Return: AX = FFFFh
Return: AL = 4Fh if function supported
    AH = status
    00h successful
    01h failed

```

Note: most of this info is from <http://grafi.ii.pw.edu.pl/gbm/matrox>

SeeAlso: MEM C000h:7FFCh"Matrox"

Format of Matrox-specific VESA extensions information:

Offset	Size	Description (Table 00125)
00h	7 BYTES	signature "VBE/MGA"
07h	10 BYTES	??? (01h 01h 00h 00h 00h 00h 00h 00h 00h 07h)
11h	WORD	BIOS version (high byte = major, low byte = one-digit minor)
13h	DWORD	-> manufacturer name (ASCIZ "Matrox")
17h	DWORD	-> ASCIZ card name
1Bh	DWORD	-> ??? ASCIZ string (empty string on card examined)
1Fh	DWORD	-> ASCIZ vendor name
23h	DWORD	-> ASCIZ BIOS file base name
27h	DWORD	-> PINS
2Bh	WORD	video memory size in 64K units
2Dh	DWORD	linear address of linear frame buffer
31h	DWORD	linear address of MGA control aperture
35h	WORD	PCI bus/device for card???
37h	201 BYTES	unused???, (cleared to 00h)

Format of Matrox PINS data (version 3.0):

Offset	Size	Description (Table 00126)
00h	2 BYTES	signature ".A" (2Eh 41h)
02h	BYTE	structure length (40h for v3.0)
03h	BYTE	???
04h	WORD	PINS version (high byte = major, low byte = minor)
06h	WORD	date last programmed
08h	WORD	number of times programmed

```
0Ah WORD flags
0Ch 8 BYTES ASCII serial number, three letters + five digits
14h 8 BYTES ???
1Ch 4 BYTES ASCII card version
20h DWORD PCB revision
24h BYTE maximum RAMDAC frequency in MHz (less 100)
25h BYTE maximum RAMDAC frequency???
26h BYTE maximum RAMDAC frequency???
27h BYTE ???
28h BYTE ??? timing data?
29h BYTE ???
2Ah BYTE ???
2Bh BYTE ???
2Ch BYTE ???
2Dh BYTE ???
2Eh BYTE ???
2Fh BYTE ???
30h 4 BYTES ???
34h BYTE ???
35h BYTE ???
36h BYTE ???
37h BYTE memory size in 64K units
38h BYTE ???
39h BYTE ???
3Ah BYTE ???
3Bh BYTE ???
3Ch BYTE ???
3Dh BYTE ???
3Eh BYTE ???
3Fh BYTE checksum (to make sum of all bytes MOD 256 == 00h)
-----V-104F15BL00-----
INT 10 - VESA VBE/DC (Display Data Channel) - INSTALLATION CHECK / CAPABILITIES
  AX = 4F15h
  BL = 00h
Return: AL = 4Fh if function supported
  AH = status
  00h successful
  BX = ???
  01h failed
  ???
SeeAlso: AX=4F15h/BL=01h,AX=4F15h/BL=02h
```

```
-----V-104F15BL01-----
INT 10 - VESA VBE/DC (Display Data Channel) - READ EDID
  AX = 4F15h
  BL = 01h
  CX = 0000h
  DX = 0000h
```

ES:DI -> 128-byte buffer for EDID record (see #00127)

Return: AL = 4Fh if function supported

AH = status

00h successful

ES:DI buffer filled

01h failed (e.g. non-DDC monitor)

SeeAlso: AX=4F15h/BL=00h,AX=4F15h/BL=02h,I2C A0h"DDC"

Format of VESA EDID record:

Offset Size Description (Table 00127)

00h 8 BYTEs padding (all FFh, or 00h FFh..FFh 00h)

08h WORD big-endian manufacturer ID (see #00136)

bits 14-10: first letter (01h='A', 02h='B', etc.)

bits 9-5: second letter

bits 4-0: third letter

0Ah WORD EDID ID code -- identifies monitor model

0Ch DWORD serial number or FFFFFFFFh

for "MAG", subtract 7000000 to get actual serial number

for "QQI", subtract 456150000

for "PHL", subtract ???

for "VSC", subtract 640000000

10h BYTE week number of manufacture

11h BYTE manufacture year - 1990

12h BYTE EDID version

13h BYTE EDID revision

14h BYTE video input type (see #00129)

15h BYTE maximum horizontal size in cm

16h BYTE maximum vertical size in cm

17h BYTE gamma factor (gamma = 1.0 + factor/100, so max = 3.55)

18h BYTE DPMS flags (see #00128)

19h BYTE chroma information: green X'/Y' and red X'/Y'

1Ah BYTE chroma information: white X'/Y' and blue X'/Y'

1Bh BYTE chroma information: red Y

1Ch BYTE chroma information: red X

1Dh BYTE chroma information: green Y

```

1Eh BYTE chroma information: green X
1Fh BYTE chroma information: blue Y
20h BYTE chroma information: blue X
21h BYTE chroma information: white Y
22h BYTE chroma information: white X
23h BYTE established timings 1 (see #00130)
24h BYTE established timings 2 (see #00131)
25h BYTE manufacturer's reserved timing or 00h for none
    bit 7: 1152x870 @ 75 Hz (Mac II, Apple)
26h 8 WORDs standard timing identification
    resolution (low byte) and vertical frequency (high byte) for
    each of eight modes (see #00132)
    X resolution = (lowbyte + 31) * 8
36h 18 BYTES detailed timing description #1 (see #00133)
    (v1.1) alternately, text identification (see #00135)
48h 18 BYTES detailed timing description #2
    (v1.1) alternately, text identification (see #00135)
5Ah 18 BYTES detailed timing description #3
    (v1.1) alternately, text identification (see #00135)
6Ch 18 BYTES detailed timing description #4
    (v1.1) alternately, text identification (see #00135)
7Eh BYTE unused???
7Fh BYTE checksum
    (radix-complement: 256-low byte of 16-bit sum of 00h-7Eh)

```

Bitfields for DPMS EDID flags:

Bit(s) Description (Table 00128)

```

2-0 unused???
3 display type
    =0 non-RGB multicolor
    =1 RGB color
4 unused???
5 Active Off supported
6 Suspend supported
7 Standby supported

```

SeeAlso: #00127

Bitfields for DPMS input signal type:

Bit(s) Description (Table 00129)

```

0 separate sync
1 composite sync

```

2 sync on green
4-3 unused???
6-5 voltage level
00 0.700V/0.300V (1.00 Vp-p)
01 0.714V/0.286V
10 0.100V/0.400V
11 reserved
7 =1 digital signal, =0 analog

SeeAlso: #00127

Bitfields for DPMS established timings 1:

Bit(s) Description (Table 00130)

0 720x400 @ 70 Hz (VGA 640x400, IBM)
1 720x400 @ 88 Hz (XGA2)
2 640x480 @ 60 Hz (VGA)
3 640x480 @ 67 Hz (Mac II, Apple)
4 640x480 @ 72 Hz (VESA)
5 640x480 @ 75 Hz (VESA)
6 800x600 @ 56 Hz (VESA)
7 800x600 @ 60 Hz (VESA)

SeeAlso: #00127,#00131

Bitfields for DPMS established timings 2:

Bit(s) Description (Table 00131)

0 800x600 @ 72 Hz (VESA)
1 800x600 @ 75 Hz (VESA)
2 832x624 @ 75 Hz (Mac II)
3 1024x768 @ 87 Hz interlaced (8514A)
4 1024x768 @ 60 Hz (VESA)
5 1024x768 @ 70 Hz (VESA)
6 1024x768 @ 75 Hz (VESA)
7 1280x1024 @ 75 Hz (VESA)

SeeAlso: #00127,#00130

Bitfields for DPMS standard timing information:

Bit(s) Description (Table 00132)

5-0 vertical refresh frequency - 60 (Hz)
7-6 aspect ratio (Y resolution = X resolution * aspect ratio)
00 ???
01 0.75
10 0.8

11 0.5625

Note: if both bytes of the timing are 00h or 01h, then the Standard Timing is "None"

SeeAlso: #00127

Format of Detailed Timing Description:

Offset Size Description (Table 00133)

00h	BYTE	horizontal frequency in kHz (if 00h, may be text) (see #00135)
01h	BYTE	vertical frequency in Hz
02h	BYTE	horizontal active time (pixels) and X resolution
03h	BYTE	horizontal blanking time (pixels)
04h	BYTE	horizontal active time 2 / horizontal blanking time 2
05h	BYTE	vertical active time (lines) and Y resolution
06h	BYTE	vertical blanking time (lines)
07h	BYTE	vertical active time 2 / vertical blanking time 2
08h	BYTE	horizontal sync offset (pixels)
09h	BYTE	horizontal sync pulsewidth (pixels)
0Ah	BYTE	vertical sync offset / vertical sync pulsewidth
0Bh	BYTE	vertical/horizontal sync offset 2 / vert/hor. sync pulsewidth 2
0Ch	BYTE	horizontal image size (mm)
0Dh	BYTE	vertical image size (mm)
0Eh	BYTE	horizontal image size 2 / vertical image size 2
0Fh	BYTE	horizontal border (pixels)
10h	BYTE	vertical border (lines)
11h	BYTE	type of display (see #00134)

SeeAlso: #00127, #00135

Bitfields for EDID detailed display type:

Bit(s) Description (Table 00134)

7	interlaced
6-5	stereo mode
00	normal display (no stereo)
01	stereo, right stereo sync high
10	stereo, left stereo sync high
11	undefined
4-3	sync type
00	sync analog composite
01	sync bipolar analog composite
10	sync digital composite
11	sync digital separate
---sync digital separate---	


```
2 vertical sync polarity (0 = negative, 1 = positive)
1 horizontal sync polarity (0 = negative, 1 = positive)
---other sync types---
2 serrate
1 sync location (0 = on green, 1 = on RGB)
-----
0 not used???
SeeAlso: #00133
```

Format of EDID Text Identification Strings:

```
Offset Size Description (Table 00135)
00h 3 BYTES 00h 00h 00h (to distinguish from detailed timing description)
03h BYTE text identifier
    FFh serial number
    FEh vendor name
    FDh vertical/horizontal frequency range
    FCh model name
---frequency range---
04h BYTE 00h ???
05h BYTE minimum vertical refresh frequency in Hz
06h BYTE maximum vertical refresh
07h BYTE minimum horizontal frequency in kHz
08h BYTE maximum horizontal frequency
09h BYTE FFh ???
---other---
04h 14 BYTES text, may be terminated with either a NUL (00h) or LF (0Ah)
SeeAlso: #00132
```

(Table 00136)

Values for manufacturer ID:

```
"ACT" Targa
"ADI" ADI Corporation http://www.adi.com.tw
"AOC" AOC International (USA) Ltd. (see #00137)
"API" Acer America Corp. (see #00138)
"APP" Apple Computer, Inc. (see #00139)
"ART" ArtMedia
"AST" AST Research (see #00140)
"CPL" Compal Electronics, Inc. / ALFA (see #00141)
"CPQ" COMPAQ Computer Corp. (see #00142)
"CTX" CTX - Chuntex Electronic Co. (see #00143)
"DEC" Digital Equipment Corporation (see #00144)
```

"DEL" Dell Computer Corp. (see #00145)
"DPC" Delta Electronics, Inc. (see #00146)
"DWE" Daewoo Telecom Ltd (see #00147)
"ECS" ELITEGROUP Computer Systems (see #00148)
"EIZ" EIZO (see #00149)
"FCM" Funai Electric Company of Taiwan
"GSM" LG Electronics Inc. (GoldStar Technology, Inc.) (see #00158)
"GWY" Gateway 2000
"HEI" Hyundai Electronics Industries Co., Ltd. (see #00150)
"HIT" Hitachi (see #00151)
"HSL" Hansol Electronics (see #00152)
"HTC" Hitachi Ltd. / Nissei Sangyo America Ltd. (see #00153)
"HWP" Hewlett Packard (see #00154)
"IBM" IBM PC Company (see #00155)
"ICL" Fujitsu ICL (see #00156)
"IVM" Idek Iiyama North America, Inc. (see #00157)
"KFC" KFC Computek
"LKM" ADLAS / AZALEA
"LNK" LINK Technologies, Inc.
"LTN" Lite-On
"MAG" MAG InnoVision (see #00159)
"MAX" Maxdata Computer GmbH
"MEI" Panasonic Comm. & Systems Co. (see #00168)
"MEL" Mitsubishi Electronics (see #00160)
"MIR" miro Computer Products AG (see #00161)
"MTC" MITAC
"NAN" NANA0 (see #00162)
"NEC" NEC Technologies, Inc. (see #00163)
"NOK" Nokia (see #00164)
"OQI" OPTIQUEST (see #00165)
"PBN" Packard Bell
"PGS" Princeton Graphic Systems (see #00166)
"PHL" Philips Consumer Electronics Co. (see #00167)
"REL" Relisys
"SAM" Samsung (see #00169)
"SDI" Samtron (see #00170)
"SMI" Smile
"SNI" Siemens Nixdorf
"SNY" Sony Corporation
"SPT" Sceptre
"SRC" Shamrock Technology

"STP" Sceptre
"TAT" Tatung Co. of America, Inc. (see #00171)
"TRL" Royal Information Company (see #00172)
"TSB" Toshiba, Inc.
"UNM" Unisys Corporation
"VSC" ViewSonic Corporation (see #00173)
"WTC" Wen Technology
"ZCM" Zenith Data Systems
"___" Targa

SeeAlso: #00127

(Table 00137)

Values for AOC monitor EDID:

A566h AOC SPECTRUM 5N1r
A782h AOC SPECTRUM 7N1r
D350h AOC SPECTRUM 4V, 4VA, 4V1r, 4V1rA
D566h AOC SPECTRUM 5V1r, 5V1rA
E570h AOC SPECTRUM 5G1r
E750h AOC SPECTRUM 7D1rA

SeeAlso: #00136,#00138

(Table 00138)

Values for Acer monitor EDID:

0037h AcerView 55
424Ch AcerView 76ie
440Bh AcerView 11D
4421h AcerView 33D
4522h AcerView 7134e
4536h AcerView 7154e
4538h AcerView 7156e
454Ch AcerView 7176ie
454Eh AcerView 7178ie
4938h AcerView 7156i
494Ch AcerView 76i
494Eh AcerView 78i
4962h AcerView 98i
4C21h AcerView 33DL
4C22h AcerView 34TL
4C37h AcerView 55L
4C38h AcerView 56L
4E4Ch AcerView 76N

5321h AcerView 7133s
5322h AcerView 7134s
5336h AcerView 7154s
5338h AcerView 7156s, 7156is
534Ch AcerView 7176is
5422h AcerView 34T
SeeAlso: #00136,#00137,#00140

(Table 00139)

Values for Apple Computer monitor EDID:

0352h AppleVision 850

SeeAlso: #00136

(Table 00140)

Values for AST monitor EDID:

8001h ASTVision 4V

8002h ASTVision 4I

8003h ASTVision 4N

8004h ASTVision 4L

8005h ASTVision 5L

8006h ASTVision 7L

8007h ASTVision 7H

8008h ASTVision 20H

8009h AST Sabre

800Ah ASTVision 5V

SeeAlso: #00136,#00127,#00138,#00141

(Table 00141)

Values for ALFA monitor EDID:

096Bh ALFA TH-450

SeeAlso: #00136,#00140,#00142

(Table 00142)

Values for Compaq monitor EDID:

0011h COMPAQ 1024

0012h COMPAQ 1024

0013h COMPAQ 1024

0014h COMPAQ 1024

0015h COMPAQ 1024

0016h COMPAQ 1024

0017h COMPAQ PRB

0018h COMPAQ PRB
0019h COMPAQ PRB
001Ah COMPAQ PRB
0020h COMPAQ PRB
0021h COMPAQ PRB
0022h COMPAQ 151FS
0023h COMPAQ 151FS
0024h COMPAQ 151FS
0025h COMPAQ 151FS
0026h COMPAQ 151FS
0027h COMPAQ 151FS
0028h COMPAQ 151FS
0029h COMPAQ PRB
002Ah COMPAQ PRB
002Bh COMPAQ PRB
002Ch COMPAQ PRB
002Dh COMPAQ 171FS
002Eh COMPAQ 171FS
002Fh COMPAQ 171FS
0030h COMPAQ 171FS
0031h COMPAQ 171FS
0032h COMPAQ 171FS
0033h COMPAQ 171FS
0040h COMPAQ QVision 200
0041h COMPAQ QVision 200
0042h COMPAQ QVision 200
0043h COMPAQ QVision 200
0044h COMPAQ QVision 200
0045h COMPAQ QVision 200
0046h COMPAQ QVision 200
0047h COMPAQ QVision 200
0048h COMPAQ QVision 200
0049h COMPAQ QVision 200
0100h COMPAQ 1024
0146h COMPAQ 1024
0147h COMPAQ 1024
0346h COMPAQ 151FS
0347h COMPAQ 151FS
0348h COMPAQ 151FS
0349h COMPAQ 151FS
034Ah COMPAQ 151FS

034Bh COMPAQ 151FS
0546h COMPAQ 171FS
0547h COMPAQ 171FS
0548h COMPAQ 171FS
0549h COMPAQ 171FS
0746h COMPAQ PRB
0747h COMPAQ PRB
0846h COMPAQ PRA
0847h COMPAQ PRA
0848h COMPAQ PRA
0849h COMPAQ PRA
084Ah COMPAQ PRA
084Bh COMPAQ PRA
084Ch COMPAQ PRA
084Dh COMPAQ PRA
0946h COMPAQ PRB
0947h COMPAQ PRB
0948h COMPAQ PRB
0949h COMPAQ PRB
0A46h COMPAQ PRA
0A47h COMPAQ PRA
0A48h COMPAQ PRA
0A49h COMPAQ PRA
0A4Ah COMPAQ PRA
0A4Bh COMPAQ PRA
0A4Ch COMPAQ PRA
0A4Dh COMPAQ PRA
0D46h COMPAQ 140
0D47h COMPAQ 140
0D48h COMPAQ 140
0D49h COMPAQ 140
0D4Ah COMPAQ 140
0D4Bh COMPAQ 140
0D4Ch COMPAQ 140
0D4Dh COMPAQ 140
0F46h COMPAQ 150
0F47h COMPAQ 150
0F48h COMPAQ 150
0F49h COMPAQ 150
0F4Ah COMPAQ 150
0F4Bh COMPAQ 150

0F4Ch COMPAQ 150
0F4Dh COMPAQ 150
SeeAlso: #00136,#00141,#00143

(Table 00143)

Values for "CTX" monitor EDID:

1451h CTX 1451
1551h CTX 1551
1562h CTX 1562
1565h CTX 1565
1569h CTX 1569S 15-inch
1765h CTX 1765
1785h CTX 1785 XE
2085h CTX 2085
2185h CTX 2185

SeeAlso: #00136,#00142,#00144

(Table 00144)

Values for Digital Equipment Corporation monitor EDID:

06FAh Digital 21" Color (FR-PCXAV-WZ)
073Ah Digital 17" Color (FR-PCXAV-YZ)
BA08h Digital 15" Color Monitor (FR-PCXBV-E*)
DA08h Digital 17" Color Monitor (FR-PCXBV-F*)

SeeAlso: #00136,#00143,#00145

(Table 00145)

Values for Dell monitor EDID:

139Ah Dell Ultrascan 14XE
139Bh Dell Ultrascan 14XE
139Ch Dell Ultrascan 14XE
139Dh Dell Ultrascan 14XE
2210h Dell Ultrascan V17X
2211h Dell Ultrascan V17X
2212h Dell Ultrascan V17X
2213h Dell Ultrascan V17X
2214h Dell Ultrascan 21TE
2215h Dell Ultrascan 21TE
2216h Dell Ultrascan 21TE
2217h Dell Ultrascan 21TE
3024h Dell VS17X
3025h Dell VS17X

3026h Dell VS17X
3027h Dell VS17X
4273h Dell VS15X

SeeAlso: #00136,#00144,#00146

(Table 00146)

Values for Delta Electronics monitor EDID:

0456h Delta DA-456
1565h Delta DA-1565
1765h Delta DA-1765

SeeAlso: #00136,#00145,#00147

(Table 00147)

Values for Daewoo monitor EDID:

1423h Daewoo CMC-1423B1
1427h Daewoo CMC-1427X1
1502h Daewoo CMC-1502B1
1505h Daewoo CMC-1505X
1507h Daewoo CMC-1507X1
1703h Daewoo CMC-1703B
5133h Daewoo CMC-1511B
7044h Daewoo CMC-1704C

SeeAlso: #00136,#00146,#00148

(Table 00148)

Values for ELITEGROUP Computer Systems monitor EDID:

0001h ECS VERTOS 1401
0002h ECS VERTOS 1501
0003h ECS VERTOS 1502
0004h ECS VERTOS 1503
0005h ECS VERTOS 1700
0006h ECS VERTOS 1701
0007h ECS VERTOS 1700
0008h ECS VERTOS 2101
0009h ECS VERTOS 2102

SeeAlso: #00136,#00147,#00149

(Table 00149)

Values for EIZO monitor EDID:

0200h EIZO FlexScan F351
0201h EIZO FlexScan F553

0202h EIZO FlexScan F563
0203h EIZO FlexScan F764
0204h EIZO FlexScan F784
0206h EIZO FlexScan 6600
0300h EIZO FlexScan 6500
0302h EIZO 9060S
0303h EIZO 9065S
0305h EIZO FlexScan T563
0306h EIZO 9070S
0307h EIZO 9080i
0308h EIZO 9400i
0309h EIZO 9500
030Ah EIZO FlexScan F340iW
030Ch EIZO FlexScan F552
030Dh EIZO FlexScan F560iW
030Fh EIZO FlexScan F760iW
0310h EIZO FlexScan F780iW
0311h EIZO FlexScan T560i
0312h EIZO FlexScan T660i
0313h EIZO FlexScan T562
0314h EIZO FlexScan T662
038Ch EIZO FlexScan F550iW
SeeAlso: #00136,#00148,#00150

(Table 00150)

Values for Hyundai monitor EDID:

0B42h Hyundai Deluxscan 21
12F0h Hyundai Deluxscan 14S
16D8h Hyundai Deluxscan 15B
16E8h Hyundai Deluxscan 15G
16EEh Hyundai Delucscan 15G+
1E02h Hyundai Deluxscan 17 Pro
1EB8h Hyundai Deluxscan 17B
1EBEh Hyundai Deluxscan 17B+
5864h Hyundai DeluxScan 15 Pro
B81Eh Hyundai DeluxScan 17
SeeAlso: #00136,#00149,#00151

(Table 00151)

Values for Hitachi monitor EDID:

1717h Hitachi Accuvue GX17L

1727h Hitachi HM1764
1827h Hitachi HM1782
2147h Hitachi HM4721
2149h Hitachi HM4921
4020h Hitachi HM4020
4021h Hitachi HM4021
4711h Hitachi Accuvue UX4721
4810h Hitachi Accuvue GX20
4811h Hitachi Accuvue GX21
4820h Hitachi HM4820
4821h Hitachi HM4821
4830h Hitachi Accuvue GX20H
4911h Hitachi Accuvue UX4921
6421h Hitachi HM6421
6811h Hitachi Accuvue UX6821
6821h Hitachi HM6821

SeeAlso: #00136,#00150,#00152

(Table 00152)

Values for Hansol Electronics monitor EDID:

0579 Hansol Electronics Mazellan14px
057A Hansol Electronics Mazellan400A
05DD Hansol Electronics Mazellan15ax
A605 Hansol Electronics Mazellan17px

SeeAlso: #00136,#00151,#00153

(Table 00153)

Values for Hitachi, Ltd. monitor EDID:

AB6Fh CM-1711M
ABC2h CM-2112M
ABC7h CM-2111M
ABCCh CM-2110M
ABE0h Hitachi, Ltd CM802
ABE2h Hitachi, Ltd CM801
ABE3h Hitachi, Ltd CM800
ABEAh Hitachi, Ltd CM803
ABF4h Hitachi, Ltd CM701
AFC8h Hitachi, Ltd CM500
AFD2h Hitachi, Ltd CM600
AFD7h Hitachi, Ltd CM611

SeeAlso: #00136,#00152,#00154

(Table 00154)

Values for Hewlett-Packard monitor EDID:

0AF0h HP D2800A Ultra VGA 1600 21"
0AF6h HP D2806A Ergo Ultra VGA 15"
0AF8h HP D2808 1024 Low Emission
0AFFh HP D2815A 1024 Low Emission 14"
0F11h HP D3857A Multi Media 15"
0F12h HP D3858A Multi Media 14"

SeeAlso: #00136,#00151,#00155

(Table 00155)

Values for IBM monitor EDID:

198Eh IBM G41
198Fh IBM G50
1990h IBM G70
1991h IBM G200
1999h IBM P50
199Ah IBM P70
199Bh IBM P200
1BB7h IBM 7095
1BB9h IBM 7097
2112h IBM 2112
2113h IBM 2113
2114h IBM 2114
2115h IBM 2115
2117h IBM 2117
2215h IBM 2215
2238h IBM 2238
2248h IBM 2248
2264h IBM 2264
2535h IBM 9525-0X1
26ACh IBM Aptiva 9900
27ADh IBM Aptiva 9901
6312h IBM 6312
6314h IBM 6314
6315h IBM 6315
6317h IBM 6317
6319h IBM 6319
6321h IBM 6321
6322h IBM 6322

6324h IBM 6324
6325h IBM 6325
6327h IBM 6327
8504h IBM 8504
8511h IBM 8511
8512h IBM 8512
8513h IBM 8513
8514h IBM 8514
8515h IBM 8515
8517h IBM 8517
8518h IBM 8518
9504h IBM 9504
9515h IBM 9515
9517h IBM 9517
9518h IBM 9518
9521h IBM 9521
9524h IBM 9524
9525h IBM 9525
9527h IBM 9527

SeeAlso: #00136,#00154,#00156

(Table 00156)

Values for Fujitsu ICL monitor EDID:

0100h Fujitsu ICL ErgoPro 211v
0200h Fujitsu ICL ErgoPro 171p
0400h Fujitsu ICL ErgoPro 171v
0700h Fujitsu ICL ErgoPro 151p
0800h Fujitsu ICL ErgoPro 151p AutoBrite
0A00h Fujitsu ICL ErgoPro 151v
0B00h Fujitsu ICL ErgoPro 141v
0D00h Fujitsu ICL ErgoPro 141p
0F00h Fujitsu ICL ErgoPro 152v
1400h Fujitsu ICL ErgoPro 142v
1600h Fujitsu ICL ErgoPro e153
1900h Fujitsu ICL ErgoPro x173
1A00h Fujitsu ICL ErgoPro x173a
1C00h Fujitsu ICL ErgoPro x152
1D00h Fujitsu ICL ErgoPro e173
2200h FUJITSU ErgoPro e154
2400h FUJITSU ErgoPro x174

SeeAlso: #00136,#00155,#00157

(Table 00157)

Values for Idek Iiyama North America monitor EDID:

1700h Iiyama Vision Master MF-8617

SeeAlso: #00136,#00156,#00158

(Table 00158)

Values for LG Electronics monitor EDID:

36B4h LG StudioWorks 44m

36B9h LG StudioWorks 44i

3AA0h LG 1505s

3AA2h LG StudioWorks 56m

3AAFh LG StudioWorks 56T 15-inch

3AB0h LG StudioWorks 5D

3AB6h LG StudioWorks 5D

426Ch LG StudioWorks 78i

426Dh LG StudioWorks 78T

426Eh LG StudioWorks 76i

4273h LG StudioWorks 76m

4274h LG StudioWorks 78m

4277h LG StudioWorks 74m

4278h LG StudioWorks 74i

427Eh LG StudioWorks 7D

427Fh LG StudioWorks 78D

4280h LG StudioWorks 78DT

4281h LG StudioWorks 7DT

4284h LG StudioWorks 76T

42CFh LG 1725s

4E21h LG StudioWorks 20i

SeeAlso: #00136,#00157,#00159

(Table 00159)

Values for MAG monitor EDID:

5620h DX1595

5624h DX15T

5626h DX1795

SeeAlso: #00136,#00158,#00160

(Table 00160)

Values for Mitsubishi monitor EDID:

0040h Mitsubishi Diamond Pro 21TX (THN-9105)

2040h Mitsubishi Diamond Scan 20H (FR-8905)
2040h Mitsubishi Diamond Pro 20X (FR-8905B)
6140h Mitsubishi Diamond Scan 17HX (FFF8705)
8040h Mitsubishi Diamond Pro 15H (SD-57xxC)
4040h Mitsubishi Diamond Pro 17TX (TFG-8705)
408Fh Mitsubishi Diamond Scan 15VX (SD58xx)
40C0h Mitsubishi Diamond Pro 87TXM (TFM8705)

SeeAlso: #00136,#00159,#00161

(Table 00161)

Values for Miro monitor EDID:

0721h PROOFSCREEN miroC21107
6815h miroD1568
6817h PROOFSCREEN miroC1768
6917h miroD1769
8217h PROOFSCREEN miroC1782
8520h PROOFSCREEN miroC2085 E
8521h PROOFSCREEN miroC2185
9321h PROOFSCREEN miroC2193

SeeAlso: #00136,#00160,#00162

(Table 00162)

Values for Nanao monitor EDID:

0000h NANA O USA FlexScan T2-20
0400h NANA O USA F2-15
0401h NANA O USA F2-17
0402h NANA O USA F2-17EX
0403h NANA O USA F2-21
0404h NANA O USA FX2-21
0405h NANA O USA FlexScan T2-17TS
0406h NANA O FlexScan 6600
0502h NANA O 9060S
0503h NANA O 9065S
0506h NANA O 9070U
0507h NANA O 9080i
0508h NANA O 9400i
0509h NANA O 9500
050Ah NANA O FlexScan F340iW
050Bh NANA O FlexScan F550i
050Ch NANA O FlexScan F550iW
050Dh NANA O FlexScan F560iW

050Eh NANA FlexScan F750i
050Fh NANA FlexScan F760iW
0510h NANA FlexScan F780iW
0511h NANA FlexScan T560i
0512h NANA FlexScan T660i
0513h NANA USA FlexScan T2-17
0514h NANA USA FlexScan T2-20
0580h NANA USA FlexScan 6300
0800h NANA FlexScan 33F
0802h NANA FlexScan 88F
0805h NANA FlexScan 54T
0882h NANA FlexScan 52F
0900h NANA FlexScan 6500
0902h NANA 9060S
0906h NANA 9070U
0907h NANA 9080i
090Ah NANA FlexScan F347
090Bh NANA FlexScan F550i
090Ch NANA FlexScan F557
090Fh NANA FlexScan 77F
0910h NANA FlexScan F780iJ
0911h NANA FlexScan T560iJ
0912h NANA FlexScan T660iJ
0913h NANA FlexScan 56T
0914h NANA FlexScan 68T
098Ah NANA FlexScan F347II
098Ch NANA FlexScan 55F
098Fh NANA FlexScan 76F
0991h NANA FlexScan T567
0993h NANA FlexScan 53T
SeeAlso: #00136,#00161,#00163

(Table 00163)

Values for NEC monitor EDID:

37FAh NEC MultiSync XV14
37FBh NEC MultiSync XV14
3A66h NEC MultiSync C400
3C00h NEC MultiSync XE15
3C0Ah NEC MultiSync XP15
3C14h NEC MultiSync XV15
3C1Eh NEC MultiSync XE15

3D5Eh NEC MultiSync XV15+
3D68h NEC MultiSync M500
3E4Eh NEC MultiSync C500
43A8h NEC MultiSync XE17
43B2h NEC MultiSync XE17
43BCh NEC MultiSync XV17
4416h NEC MultiSync XP17
533Eh NEC MultiSync XE21
53B6h NEC MultiSync XP21

SeeAlso: #00136,#00162,#00164

(Table 00164)

Values for Nokia monitor EDID:

008Dh NOKIA 449M
0098h NOKIA 449X
00A3h NOKIA 447KC
00A9h NOKIA 447XI
00ABh NOKIA 447KA
00ADh NOKIA 447M
00B7h NOKIA 447W
00B8h NOKIA 447X
00BBh NOKIA 447XAV
00BCh NOKIA 447K
00D2h NOKIA 445R

SeeAlso: #00136,#00163,#00165

(Table 00165)

Values for OptiQuest monitor EDID:

3138h OPTIQUEST VA656
3141h OPTIQUEST 14ES
3232h OPTIQUEST 1562A-2
3233h OPTIQUEST 1769DC
3234h OPTIQUEST 1000S-2
3332h OPTIQUEST V655
3333h OPTIQUEST V775
3432h OPTIQUEST V650
3832h OPTIQUEST V665
4637h OPTIQUEST 1769DC

SeeAlso: #00136,#00164,#00166

(Table 00166)

Values for Princeton Graphics monitor EDID:

003Bh Princeton E015

003Dh Princeton E017

SeeAlso: #00136,#00165,#00167

(Table 00167)

Values for Philips monitor EDID:

0200h Philips CM0200 (15C)

0500h Philips CM0500 (20C)

0700h Philips CM0700 (20T)

0800h Philips CM0800 (15B)

1200h Philips CM1200 (15A)

1800h Philips CM1800 (15A)

2000h Philips CM0200 (14B)

2600h Philips 17TCM26

2800h Philips 17BCM28

3800h Philips 17ACM38

5600h Philips CM5600 (20B)

700Bh Philips CM0700 (21B)

8000h Philips CM0800 (14A)

SeeAlso: #00136,#00166,#00168

(Table 00168)

Values for Panasonic monitor EDID:

1604h Panasonic TX-D2151W-ES

1610h Panasonic TX-D2151NM

SeeAlso: #00136,#00167,#00169

(Table 00169)

Values for Samsung monitor EDID:

0000h Samsung SyncMaster 3Ne

0100h Samsung SyncMaster 4S

1530h Samsung 15GLsi

4610h Samsung SyncMaster 21GLs

4690h Samsung SyncMaster 20GLsi

4D50h Samsung SyncMaster 15GLE

4D51h Samsung SyncMaster 15GLi

4D52h Samsung SyncMaster 15M

4D70h Samsung SyncMaster 17GLi

4D71h Samsung SyncMaster 17GLsi

4D72h Samsung SyncMaster 6Ne

4D73h Samsung 17GLi
4D74h Samsung 17GLsi
5450h Samsung SyncMaster 15Me
6D20h Samsung SyncMaster 15GLE

SeeAlso: #00136,#00168,#00170

(Table 00170)

Values for Samtron monitor EDID:

1428h Samtron 428PT/PTL
1528h Samtron SC-528TXL
1529h Samtron SC-528UXL
1530h Samtron SC-528MXLJ
4690h Samtron SC-208DXL+
4D70h Samtron SC-728FXL
4D71h Samtron SC-726GXL
5451h Samtron SC-528MDL

SeeAlso: #00136,#00169,#00171

(Table 00171)

Values for Tatung monitor EDID:

1F65h Intelliscan TM651x series 15"
1F67h Intelliscan TM671x series 17"
2F44h Intelliscan TM442x series 14"
2F45h Intelliscan TM452x series 15"
434Dh Tatung CM-17MC
4855h Tatung CM-14UH

SeeAlso: #00136,#00170,#00172

(Table 00172)

Values for Royal Information Company monitor EDID:

061Ch TRL/RIC DL-1564
0622h TRL/RIC DH-1570

SeeAlso: #00136,#00171,#00173

(Table 00173)

Values for ViewSonic monitor EDID:

0C00h ViewSonic 17GS
0C0Fh ViewSonic 17PS
0C1Fh ViewSonic 17GA
1600h ViewSonic 21PS-2
2601h ViewSonic 15GS

2600h ViewSonic 15GS-2
3141h ViewSonic 14ES
3142h ViewSonic 14ES
3143h ViewSonic 14ES
3145h ViewSonic 15GS-3
3252h ViewSonic 21PS-2
3253h ViewSonic 21PS-2
334Bh ViewSonic 17GS-2
334Ch ViewSonic 17GS-2
3351h ViewSonic PT810
344Bh ViewSonic 17PS-2
344Ch ViewSonic 17PS-2
3451h ViewSonic PT810-2
354Dh ViewSonic GT800
354Eh ViewSonic GT800
3550h ViewSonic GT800
3644h ViewSonic 15GA
3646h ViewSonic 15GA
3648h ViewSonic 15GA
364Ah ViewSonic PT770
364Bh ViewSonic PT770
364Ch ViewSonic PT770
3744h ViewSonic 15G-2
3746h ViewSonic 15G-2
3748h ViewSonic 15G-2
374Bh ViewSonic 17GA
374Ch ViewSonic 17GA
3844h ViewSonic 15ES-2
3846h ViewSonic 15ES-2
3848h ViewSonic 15ES-2
384Ah ViewSonic 17EA
384Bh ViewSonic 17EA
384Ch ViewSonic 17EA
3944h ViewSonic 15GS-2
3946h ViewSonic 15GS-2
3948h ViewSonic 15GS-2
394Ah ViewSonic 17GS-2
394Bh ViewSonic 17GS-2
394Ch ViewSonic 17GS-2
424Ah ViewSonic GT770
424Bh ViewSonic GT770

424Ch ViewSonic GT770
434Ah ViewSonic 17PS-2
434Bh ViewSonic 17PS-2
434Ch ViewSonic 17PS-2
4439h ViewSonic 15GS-2
444Ah ViewSonic 17GA-2
4637h ViewSonic 1769DC
4745h ViewSonic G653 (VCDTS21444-2E)

SeeAlso: #00136,#00172

-----V-104F15BL02-----

INT 10 - VESA VBE/DC (Display Data Channel) - READ VDIF

AX = 4F15h

BL = 02h

???

Return: AL = 4Fh if function supported

AH = status

00h successful

???

01h failed

SeeAlso: AX=4F15h/BL=00h,AX=4F15h/BL=01h

-----V-104F16-----

INT 10 - VESA VBE/GC (Graphics System Configuration) - API

AX = 4F16h

BL = function

00h installation check / get capabilities

other registers vary by function

Return: varies by function

-----V-104F17-----

INT 10 - VESA VBE/AF (Accelerator Functions) - API

AX = 4F17h

BL = function

00h installation check / get capabilities

other registers vary by function

Return: AL = 4Fh if function supported

AH = status

00h successful

01h failed

02h hardware does not support function

03h function not available in current video mode

else reserved for future error codes

other vary by function

Notes: the accelerator function code should be given a 32-bit protected mode stack which has at least 1024 bytes available for use by the VBE/AF code

when called, the I/O permission bitmap must allow access to any ports which VBE/AF may require for operation

SeeAlso: AX=4F0Bh

-----V-104F4D-----

INT 10 - VESA - VIDEO CURSOR INTERFACE REQUEST

AX = 4F4Dh

BX = number of bytes available for VCI use

DS:0000h -> buffer for VCI

ES:DI -> VCI driver callback function

Return: AL = 4Fh if supported

AH = status

00h successful

BX = number of bytes used by VCI

ES:DI -> VCI request handler

01h failed

Desc: allow the VESA BIOS Extensions to cooperate with a pointing-device (typically mouse) driver

SeeAlso: AX=4F12h

-----V-104F70-----

INT 10 - Avance Logic - GET ADAPTER INFORMATION

AX = 4F70h

Return: AX = 004Fh if successful

BL = board information (see #00174)

BH = board type???

00h on ALG2101

FFh otherwise

SeeAlso: AX=4F00h

Bitfields for Avance Logic board information:

Bit(s) Description (Table 00174)

0-1 video RAM size

00 256K

01 512K

10 1M

11 2M

2 ???

3 slot size???

=0 8-bit slot

```
=1 16-bit slot
4  unused
5  set on ALG2228/AL2301
6-7  unused
-----V-104FDD-----
INT 10 - ATI M64VBE.COM - GET RESIDENT SEGMENT
  AX = 4FDDh
Return: AX = CS of resident code
Program: M64VBE is a VESA VBE 2.0 driver TSR for ATI's Mach64 video chip
SeeAlso: AX=4FFFh/BX=364Dh"M64VBE",INT 60"M64VBE"
-----V-104FFF-----
INT 10 - VESA SuperVGA BIOS - Everex - TURN VESA ON/OFF
  AX = 4FFFh
  DL = new state (00h off, 01h on)
Return: AX = 0000h if successful
-----V-104FFF-----
INT 10 - Diamond Stealth 24 - SET/RESET DUAL DISPLAY MODE
  AX = 4FFFh
  BX = dual display mode
    00h reset
    01h set dual display, 32KB VGA test
    02h set dual display, 64KB VGA test
Return: AX = 4F00h if successful
  BX = number of scanlines off screen for test mode
-----V-104FFFBX364D-----
INT 10 - ATI M64VBE.COM - UNINSTALL
  AX = 4FFFh
  BX = 364Dh ('6M')
  CX = 5634h ('V4')
Return: AX,DX,DS,ES destroyed
Program: M64VBE is a VESA VBE 2.0 driver TSR for ATI's Mach64 video chip
SeeAlso: AX=4FDDh"M64VBE",INT 60"M64VBE"
-----U-1050-----
INT 10 - SCROLOCK.COM - INSTALLATION CHECK
  AH = 50h
Return: BX = 1954h if installed
  AL = 00h if inactive, nonzero if active
Program: SCROLOCK is a utility supplied with System Enhancement Associates'
  ARC archiver
SeeAlso: AH=51h
-----J-105000-----
```

INT 10 - VIDEO - AX PC - SET SCREEN COUNTRY CODE

AX = 5000h

BX = country code

0001h USA (English), 0051h Japan

Return: AL = status

00h successful

01h bad country code

02h other error

Note: This function is called with BX=0051h by Japanese versions of
MS-DOS/PC DOS/DR DOS IO.SYS/IBMBIO.COM at initialization time.

SeeAlso: AX=5001h, INT 16/AX=5000h

-----J-105001-----

INT 10 - VIDEO - AX PC - GET SCREEN COUNTRY CODE

AX = 5001h

Return: AL = status

00h successful

BX = country code

02h error

SeeAlso: AH=00h, AX=5000h, INT 16/AX=5001h, INT 21/AH=38h

-----V-105049-----

INT 10 - VIDEO - SCREENR v1.55+ - API

AX = 5049h ('PI')

BX = function

0000h installation check

0001h lock mode

0002h unlock mode

0003h lock palette

0004h unlock palette

Return: AX = 0000h if installed

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

CL = mode locking status

00h mode not locked

01h mode locked: INT 10/AH=00h disabled

CH = palette locking status

00h palette not locked

01h palette locked, the following functions are disabled:

AX=1000h, AX=1001h, AX=1002h, AX=1010h, AX=1012h

Program: SCREENR is a TSR supplied with Patrick Ibbetson's SCREEN display
utility.

Index: installation check;SCREENR

-----U-1051-----

INT 10 - SCROLOCK.COM - ENABLE/DISABLE

AH = 51h
AL = state
 00h disable
 nonzero enable

Return: nothing

Program: SCROLOCK is a utility supplied with System Enhancement Associates'

 ARC file archiver

SeeAlso: AH=50h"SCROLOCK"

-----J-105100-----

INT 10 - VIDEO - AX PC - REGISTER EXTERNAL CHARACTER

AX = 5100h
BH = character width in bits (10h)
BL = character height (10h)
DX = character code (DH = F0h-F3h, DL=40h-7Eh,80h-FCh)
ES:BP -> character bitmap

Return: AL = status (00h successful, 01h failed)

SeeAlso: AX=5101h,INT 1F"SYSTEM DATA"

-----J-105101-----

INT 10 - VIDEO - AX PC - READ CHARACTER

AX = 5101h
BH = character width in bits
BL = character height
DX = character code (DH = 00h if 8-bit character)
ES:BP -> buffer for character bitmap

Return: AL = status (00h successful, 01h failed)

SeeAlso: AH=09h,AX=5100h

-----J-105200-----

INT 10 - VIDEO - AX PC - SET VIRTUAL TEXT RAM BUFFER

AX = 5200h
BX = segment of buffer

Return: nothing

SeeAlso: AX=5201h

-----J-105201-----

INT 10 - VIDEO - AX PC - GET VIRTUAL TEXT RAM BUFFER

AX = 5201h
Return: BX = segment of buffer or 0000h if failed

SeeAlso: AX=8300h,AH=FEh

-----1053-----

INT 10 - Show Partner F/X v3.6 - START PRESENTATION

AH = 53h

DS:DX -> ASCIZ name of presentation file (no path, extension forced to
.PR2)

Return: ???

SeeAlso: AH=55h

-----1055-----

INT 10 - Show Partner F/X v3.6 - UNINSTALL
AH = 55h

Return: FXSHOW.EXE removed from memory

SeeAlso: AH=53h

-----V-105555-----

INT 10 - VIDEO - ATI EGA/VGA Wonder Super Switch - INSTALLATION CHECK
AX = 5555h

Return: AX = AAAAh if installed

BX:CX -> ??? routine in SMS.COM resident portion
-> data area (see #00175) in VCONFIG

Program: Super Switch (SMS.COM) is a video mode switch program supplied with
ATI EGA Wonder. It also maps video mode 08h to 27h or 23h.

SeeAlso: INT 10/AH=00h, INT 14/AX=AA01h, INT 2F/AX=6400h

Index: screen saver;ATI Wonder SMS.COM

Format of ATI Super Switch data area:

Offset Size Description (Table 00175)

00h DWORD original INT 09 vector

04h DWORD original INT 10 vector

08h DWORD original INT 1C vector

0Ch WORD screen saver state, 0=off, 1=on

0Eh WORD blanking interval in clock ticks

-----105555BXAAAA-----

INT 10 - LIR - SET CONFIGURATION

AX = 5555h

BX = AAAAh

CL = new configuration flags (see #04118)

CH > 0 new video adapter configuration (see #04121)
= 0 do not update video adapter configuration

DL = new keyboard layout flags (see #04119)

DH = new alphabet flags (see #04120)

Return: AX = A55Ah if installed

Program: LIR is a DOS/Windows 3.xx codepage support package developed by
LABAS u.a.b. It supports several screen and printer codepages as
well as various keyboard layouts used in Lithuania.

Download at: <ftp://ftp.labas.com/pub/drivers/language/lir422.zip>

SeeAlso: AX=5555h/BX=BBBBh,INT 17/AX=5555h/BX=AAAAh

Bitfields for LIR configuration flags:

Bit(s) Description (Table 04118)

- 7 program is active
- 6 ???
- 5-3 code page
- 2 Lithuanian???
- 1 Russian???
- 0 enable beeps

SeeAlso: #04119

Bitfields for LIR keyboard layout flags:

Bit(s) Description (Table 04119)

- 7-5 keyboard layout
 - 100 QWERTY
 - 010 Lithuanian AZERTY
 - 001 Lithuanian ergonomic FZVPUJ
- 4-3 reserved (0)
- 2 Lithuanian keyboard???
- 1 Russian keyboard???
- 0 ???

SeeAlso: #04118,#04120

Bitfields for LIR Alphabet flags:

Bit(s) Description (Table 04120)

- 7-3 reserved (0)
- 2 Lithuanian alphabetic symbol table???
- 1 Russian alphabetic symbol table???
- 0 reserved (0)

SeeAlso: #04118,#04119,#04121

Bitfields for LIR video adapter configuration:

Bit(s) Description (Table 04121)

- 7 Hercules Graphics Card (HGC)
- 6-5 reserved (0)
- 4 VGA
- 3 MCGA
- 2 EGA
- 1 CGA
- 0 reserved (0)

Note: multiple bits may be set if more than one video adapter is installed

SeeAlso: #04118

-----105555BXBBBB-----

INT 10 - LIR - GET CONFIGURATION

AX = 5555h

BX = BBBBh

Return: AX = magic signature A55Ah

BX = version number (BH = major version, BL = minor version),
e.g. BX = 0410h for LIR v4.10, or BX = 0422h for LIR v4.22)

CL = configuration flags (see #04118)

CH = video adapter configuration (see #04121)

DL = keyboard layout flags (see #04119)

DH = alphabet flags (see #04120)

SeeAlso: AX=5555h/BX=AAAAh, INT 17/AX=5555h/BX=BBBBh

-----V-105F00-----

INT 10 - Chips & Technologies Extended BIOS - RETURN CHIP INFORMATION

AX = 5F00h

Return: AL = 5Fh function supported

BL = chip type (see #00176)

BH = video memory size (see #00177)

CX = miscellaneous information (see #00178)

SeeAlso: AX=5F01h

Bitfields for Chips&Technologies chip type:

Bit(s) Description (Table 00176)

7-4 chip type

0000: 82c451

0001: 82c452 / 82c452A

0010: 82c455

0011: 82c453

0100: 82c450

0101: 82c456

0110: 82c457

0111: F65520

1000: F65530 / F65525

1001: F66510

1010: ???

1011: F64300 "Wingine DGX"

1100: F65535/F65545 ???

1101: F65540

1110: ???

1111: ???
3-0 revision number
SeeAlso: #00177

(Table 00177)

Values for Chips&Technologies video memory size:

00h 256KB
01h 512KB
02h 1MB
03h 2MB

SeeAlso: #00176

Bitfields for Chips&Technologies miscellaneous information:

Bit(s) Description (Table 00178)

0 DAC size (0=6-bit, 1=8-bit)
1 system type (0=PC/AT, 1=PS/2)
2 extended text modes supported by BIOS
3 reserved
4 extended graphics modes supported by BIOS
5 reserved
6 BIOS supports graphics cursor
7 BIOS supports anti-aliased font
8 BIOS supports pre-programmed emulation
9 BIOS supports auto emulation
10 BIOS supports variable mode set at cold boot
11 BIOS supports variable mode set on warm boot
12 BIOS supports emulation mode set at cold boot
13 BIOS supports emulation mode set on warm boot
14-15 reserved

-----V-105F00-----

INT 10 - VIDEO - Realtek RTVGA - RETURN CHIP VERSION

AX = 5F00h

Return: AH = 00h, if successful

AL = chip version (the same value that VTEST.EXE reads)

00h RTG3103???
01h RTG31030/RTG3105
02h RTG3106???
3Fh non-Realtek chip

BUG: in v3.C10, AX=5F00h on return due to improper stack restoration code

SeeAlso: AX=5F01h"RTVGA"

-----V-105F01-----

INT 10 - Chips & Technologies Extended BIOS - SET PREPROGRAMMED EMULATION

AX = 5F01h

BL = emulation type (see #00179)

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F00h,AX=5F02h,AH=FFh"Oak"

(Table 00179)

Values for Chips&Technologies emulation type:

00h,01h reserved

02h CGA

03h MDA

04h Hercules

05h EGA

06h VGA (disable emulation)

-----V-105F01-----

INT 10 - VIDEO - Realtek RTVGA - WRITE RTVGA BIOS STRING TO DESTINATION

AX = 5F01h

ES:DI -> zero-filled buffer for BIOS ID string

Return: AH = 00h if successful

ES:DI -> ASCII signature "REALTEK VGA BIOS Version 3C.10"

BUG: in v3.C10, AX=5F01h on return due to improper stack restoration code

SeeAlso: AX=5F00h"RTVGA",AX=5F02h"RTVGA"

-----V-105F02-----

INT 10 - Chips & Technologies Extended BIOS - AUTO EMULATION CONTROL

AX = 5F02h

BL = new state of automatic emulation (00h enabled, 01h disabled)

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F01h,AX=5F03h

-----V-105F02-----

INT 10 - Chips & Technologies '64300' BIOS - SET CLOCK

AX = 5F02h

BH = what to set

02h dot clock

BL = speed in MHz (12-99)

03h memory clock

BL = speed in MHz (12-99)
FFh default dot and memory clocks
Return: AL = 5Fh if function supported
AH = status
00h unsuccessful
01h successful
SeeAlso: AX=5F00h"Chips",AX=5F03h"64300",AX=5F02h"65530"
-----V-105F02-----
INT 10 - Chips & Technologies '65530' BIOS - SET CLOCK
AX = 5F02h
BH = what to set
02h dot clock
BL = new clock speed (see #00180)
03h memory clock
BL = new clock speed (see #00180)
04h set power-down register
BL = new value for power-down register (bits 3-0 only)
06h (404 clock chip) set control register
BL = new vlaue for control register (bits 5-0 only)
06h (404A/B/B-ES clock chips) set control register PWRDN1
BL = new value for control register
07h (404A/B/B-ES clock chips) set control register PWRDN1
BL = new value for control register
FFh default dot and memory clocks
Return: AL = 5Fh if function supported
AH = status
00h unsuccessful
01h successful
SeeAlso: AX=5F01h,AX=5F02h"64300",AX=5FA0h,#00184

(Table 00180)

Values for C&T '65530' BIOS clock speed:

00h	12 MHz
01h	14 MHz
02h	16 MHz
03h	18 MHz
04h	20 MHz
05h	25 MHz
06h	28 MHz
07h	32 MHz
08h	36 MHz

09h 40 MHz
0Ah 45 MHz
0Bh 50 MHz
0Ch 56 MHz
0Dh 60 MHz
0Eh 65 MHz
0Fh 66 MHz
15h 34 MHz

-----V-105F02-----

INT 10 - VIDEO - Realtek RTVGA - RETURN RTVGA ON-BOARD MEMORY SIZE

AX = 5F02h

Return: AH = 00h, if successful

AL = on-board memory size

(00h = 256K, 01h = 512K, 02h = 768K, 03h = 1024K)

BUG: in v3.C10, AX=5F02h on return due to improper stack restoration code

SeeAlso: AX=5F01h"RTVGA",AX=5F03h"RTVGA"

-----V-105F03-----

INT 10 - VIDEO - Realtek RTVGA - SET ???

AX = 5F03h

BL = 0-3 (???)

Return: AH = 00h if successful

BUG: in v3.C10, AX=5F03h on return due to improper stack restoration code

SeeAlso: AX=5F02h"RTVGA"

-----V-105F03BL00-----

INT 10 - Chips & Technologies Extended BIOS - SET POWER-ON DISPLAY MODE

AX = 5F03h

BL = 00h

CL = display mode

CH = mode (see #00181)

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F02h,AX=5F03h/BL=01h"Chips",AX=5F90h

Bitfields for Chips&Technologies display mode:

Bit(s) Description (Table 00181)

1-0 scanlines (00 = 200, 01 = 350, 10 = 400)

7 persistence (0 reset on next boot, 1 until changed)

-----V-105F03BH00-----

INT 10 - Chips & Technologies '64300' BIOS - GET CLOCK SPEED

```
AX = 5F03h
BH = 00h (get default memory clock)
Return: AL = 5Fh if function supported
AH = status
00h unsuccessful
01h successful
BL = clock speed in MHz
```

SeeAlso: AX=5F02h"64300",AX=5F04h"64300"

-----V-105F03BL01-----

INT 10 - Chips & Technologies Extended BIOS - SET POWER-ON EMULATION MODE

```
AX = 5F03h
BL = 01h
CL = emulation mode (see #00181)
CH = permanence
bit 7 persistence (0 reset on next boot, 1 until changed)
```

Return: AL = 5Fh if function supported

```
AH = status
00h unsuccessful
01h successful
```

SeeAlso: AX=5F02h,AX=5F03h/BL=00h"Chips",AX=5F90h

-----V-105F04-----

INT 10 - Chips & Technologies '64300' BIOS - GET REFRESH RATE

```
AX = 5F04h
BL = video mode number
Return: AL = 5Fh if function supported
AH = status
00h unsuccessful
01h successful
BH = available refresh rates (see #00182)
BL = currently-set refresh rate (see #00182)
```

SeeAlso: AX=5F03h"64300",AX=5F05h"64300"

Bitfields for C&T 64300 refresh rates:

Bit(s) Description (Table 00182)

```
5 75 Hz
4 72 Hz
3 70 Hz
2 60 Hz
1 56 Hz
0 interlaced
```

-----V-105F05-----

INT 10 - Chips & Technologies '64300' BIOS - SET REFRESH RATE

AX = 5F05h

BL = video mode number

BH = refresh rate to set (see #00182)

bit 7: =0 program new clock and CRT parametes

=1 keep current parameters

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F04h"64300"

-----V-105F06-----

INT 10 - Chips & Technologies '64300' BIOS - GET XRAM INFORMATION

AX = 5F06h

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

BL = XRAM presence (00h no XRAM, 01h present)

SeeAlso: AX=5F05h,AX=5F08h

-----V-105F08-----

INT 10 - Chips & Technologies '64300' BIOS - SET LINEAR MEMORY START ADDRESS

AX = 5F08h

BX = desired start address in megabytes

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F06h

-----V-105F10-----

INT 10 - Chips & Technologies '64300'/'65530' BIOS - GET LINEAR MEMORY INFO

AX = 5F10h

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

BX:CX = linear memory base address

DX = virtual screen width offset (bytes)

SI;DI = linear memory size

SeeAlso: AX=5F08h,AX=5FA0h

-----V-105F50-----

INT 10 - Chips & Technologies Extended BIOS - GET LAPTOP DISPLAY STATUS

AX = 5F50h

Return: DL = display status (see #00183)

Note: this call can be used on Toshiba laptops since ~1995 instead of

INT 42/AX=7503h which is no longer supported with the F655xx chips

SeeAlso: INT 42/AX=7503h

Bitfields for display status:

Bit(s) Description (Table 00183)

7-4 always 0 ???

3 1: both displays active (bit 1 also set)

2 1: normal (monochrome display only), 0: inverse or color

1 1: internal display active, 0: external only

0 always set ???

-----V-105F50-----

INT 10 - Chips & Technologies '65530' BIOS - GET 655xx INFORMATION

AX = 5F50h

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

BX = width of flat panel in pixels

CX = height of flat panel in pixels

DX = F65520/525/530 status (see #00184)

SeeAlso: AX=5F51h, INT 15/AX=5F31h

Bitfields for C&T F65520/525/530 status:

Bit(s) Description (Table 00184)

15-0 reserved

6-5 type of clock chip

00 = 404

01 = 404A

10 = 404B ES

11 = 404B

4 accelerator enabled

3 both displays active

2 video polarity (1 = inverted)

1 output device

0 CRT

1 flat panel (LCD)

0 reserved

-----V-105F51-----

INT 10 - Chips & Technologies Extended BIOS - SWITCH DISPLAY DEVICE

AX = 5F51h

BL = new active display

00h CRT

01h flat panel (LCD)

02h both simultaneously

BH = 01h to allow reprogramming the DAC

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F50h,AX=5F53h

-----V-105F53-----

INT 10 - Chips & Technologies Extended BIOS - EN/DISABLE ACCELERATOR SUPPORT

AX = 5F53h

BL = new support state

00h disable buffer/accelerator

01h enable buffer/accelerator

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F50h,AX=5F51h,AX=5F54h

-----V-105F54-----

INT 10 - Chips & Technologies Extended BIOS - TURN FLAT PANEL ON/OFF

AX = 5F54h

BL = new state

00h on

01h off

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F50h,AX=5F51h,AX=5F53h,AX=5F5Ah

-----V-105F5A-----

INT 10 - Chips & Technologies Extended BIOS - SET FLAT-PANEL VIDEO POLARITY

AX = 5F5Ah

BL = new video polarity

00h normal

01h inverted

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F50h,AX=5F54h

-----V-105F5C-----

INT 10 - Chips & Technologies Extended BIOS - SET VERTICAL COMPENSATION

AX = 5F5Ch

BL = type of vertical compensation to use

00h no compensation

01h automatic centering

02h set centering register

DX = centering value (bits 9-0 only)

03h non-automatic centering

04h set vertical line insertion register

DL = vertical line insertion (bits 3-0 only)

05h set alternate maximum scanline register

DL = alternate maximum scanline (bits 4-0 only)

06h enable text stretching, type 0

07h enable text stretching, type 1

08h enable text stretching, type 2

09h enable text stretching, type 3

0Ah disable text stretching

0Bh set vertical line replication register

DL = vertical line replication (bits 3-0 only)

0Ch enable graphics stretching, type 0

0Dh enable graphics stretching, type 1

0Eh disable vertical graphics stretching

0Fh disable all horizontal and vertical compensation

10h enable optimal compensation

11h disable optimal compensation

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F50h,AX=5F5Eh,AX=5F5Fh

-----V-105F5E-----

INT 10 - Chips & Technologies Extended BIOS - EN/DISABLE TALL FONT LOADING

AX = 5F5Eh

BL = new state of tall-font loading

00h enable loading of 8x19/30/32 fonts

01h disable loading of 8x19/30/32 fonts

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

Note: when tall fonts are enabled, the fonts are stretched by duplicating scan lines as appropriate to convert an 8x16 font to the proper height (scan lines 1/9/12 [8x19] or 2-15 [8x30] or all [8x32])

SeeAlso: AX=5F5Ch,AX=5F5Fh

-----V-105F5F-----

INT 10 - Chips & Technologies Extended BIOS - HORIZONTAL COMPENSATION

AX = 5F5Fh

BL = horizontal compensation type

00h none

01h automatic centering

02h set centering register

DL = centering value

03h non-automatic centering

04h enable text compression (force 8xN fonts)

05h disable text compression (allow 9xN fonts)

06h enable auto doubling

07h disable auto doubling

0Fh disable all horizontal and vertical compensation

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AX=5F50h,AX=5FA0h

-----V-105F90-----

INT 10 - Chips & Technologies Extended BIOS - GET SUPERVGA STATE BUFFER SIZE

AX = 5F90h

CX = state mask (see #00185)

Return: AL = 5Fh if function supported

BX = number of 64-byte blocks required

SeeAlso: AH=1Ch,AX=5F03h,AX=5F91h,AX=5F92h,AX=5FA0h

Bitfields for Chips&Technologies state mask:

Bit(s) Description (Table 00185)

0 video hardware

1 BIOS data state

2 DAC state

15 type (0 all state info, 1 SuperVGA state only)

-----V-105F91-----

INT 10 - Chips & Technologies Extended BIOS - SAVE SUPERVGA STATE

AX = 5F91h

CX = state mask (see #00185)

ES:BX -> save buffer

Return: AL = 5Fh if function supported

buffer at ES:BX filled

SeeAlso: AH=1Ch,AX=5F03h,AX=5F90h,AX=5F92h,AX=5FA1h

-----V-105F92-----

INT 10 - Chips & Technologies Extended BIOS - RESTORE SUPERVGA STATE

AX = 5F92h

CX = state mask (see #00185)

ES:BX -> previously-filled save buffer

Return: AL = 5Fh if function supported

SeeAlso: AH=1Ch,AX=5F03h,AX=5F90h,AX=5F92h,AX=5FA2h

-----V-105FA0-----

INT 10 - Chips & Technologies Wingine DGX - GET EXTENDED BIOS SAVE BUFFER SIZE

AX = 5FA0h

CX = state(s) to be saved (see #00186)

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

BX = number of 64-byte blocks required for state

SeeAlso: AH=1Ch,AX=5FA1h,AX=5FA2h,AX=5F90h,AX=4F04h

Bitfields for C&T BIOS save/restore state type:

Bit(s) Description (Table 00186)

0 video hardware

1 BIOS data

2 DAC and color registers

14 (65530 BIOS with AL=A1h only) clear emulation state

15 extended registers

SeeAlso: #00048,#00085

-----V-105FA1-----

INT 10 - Chips & Technologies Wingine DGX - SAVE VIDEO STATE

AX = 5FA1h

CX = state(s) to be saved (see #00186)

ES:BX -> buffer for state information

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

ES:BX buffer filled

SeeAlso: AH=1Ch,AX=5FA0h,AX=5FA2h,AX=5F91h

-----V-105FA2-----

INT 10 - Chips & Technologies Wingine DGX - RESTORE VIDEO STATE

AX = 5FA2h

CX = state(s) to be restored (see #00186)

ES:BX -> buffer containing previously-saved state information

Return: AL = 5Fh if function supported

AH = status

00h unsuccessful

01h successful

SeeAlso: AH=1Ch,AX=5FA0h,AX=5FA1h,AX=5F92h

-----V-1060-----

INT 10 - HUNTER 16 - SET GRAPHICS FONT

AH = 60h

AL = new font

bit 7 output characters in reverse video

bits 0-6 font number (see #00200 at AH=77h)

Return: BH = cell width

BL = cell height

Desc: select the font to use for output in mode 6; this can be used to mix fonts on the screen

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=04h"HUNTER",AH=73h"HUNTER",AH=77h"HUNTER"

-----V-1061-----

INT 10 - HUNTER 16 - MOVE CURSOR TO GRAPHICS CO-ORDINATES

AH = 61h

CX = column

DX = row

Return: nothing

Note: this function is only valid in graphics modes

SeeAlso: AH=62h

-----V-1062-----

INT 10 - HUNTER 16 - GET CURSOR POSITION IN GRAPHICS CO-ORDINATES

AH = 62h

Return: CX = column

DX = row

Note: this function is only valid in graphics modes

SeeAlso: AH=61h

-----V-1063-----

INT 10 - HUNTER 16 - SET CONTRAST

AH = 63h

AL = contrast (0..127)

Return: nothing

Desc: set the LCD screen contrast to improve visibility

SeeAlso: AH=64h,AH=75h

-----V-1064-----

INT 10 - HUNTER 16 - SET BACKLIGHT LEVEL

AH = 64h

AL = level (0..15)

Return: nothing

Desc: set the LCD backlight level to improve visibility in bad lighting

SeeAlso: AH=63h,AH=65h,AH=74h,AH=78h"HUNTER",INT 15/AH=74h

-----V-106500-----

INT 10 - HUNTER 16 - GET BACKLIGHT TIMEOUT

AX = 6500h

Return: BX = timeout in seconds (0-300)

Desc: returns the time after which the backlight will turn off

SeeAlso: AH=63h,AH=64h,AX=6501h,INT 15/AH=74h

-----V-106501-----

INT 10 - HUNTER 16 - SET BACKLIGHT TIMEOUT

AX = 6501h

BX = timeout in seconds (0-300)

Return: nothing

Desc: set the idle time after which the backlight will turn off

SeeAlso: AH=64h,AX=6500h

-----V-1066-----

INT 10 - HUNTER 16 - SPLIT DISPLAY

AH = 66h

AL = control bits (see #00187)

BH = top row of lower section of LCD (0..7)

BL = top row of lower section of the virtual screen (0..24)

Return: nothing

Desc: split the LCD screen horizontally into two parts; each part may be individually scrollable or fixed.

Note: the Hunter 16 has a 240x64 LCD display which serves as a window into a 640x200 virtual screen

Bitfields for HUNTER 16 control bits:

Bit(s) Description (Table 00187)
0-1 01: no split, 10: allow split
2 allow scrolling the upper part
3 allow scrolling the lower part

-----V-106A00BX0000-----

INT 10 - Direct Graphics Interface Standard (DGIS) - INQUIRE AVAILABLE DEVICES

AX = 6A00h
BX = 0000h
CX = 0000h
DX = buffer length (may be 0)
ES:DI -> buffer

Return: BX = number of bytes stored in buffer

CX = bytes required for all descriptions (0 if no DGIS)

Note: buffer contains descriptions and addresses of DGIS-compatible
display(s) and printer(s)

SeeAlso: AX=6A02h

-----V-106A01CX0000-----

INT 10 - Direct Graphics Interface Standard (DGIS) - REDIRECT CHARACTER OUTPUT

AX = 6A01h
CX = 0000h
ES:DI = address of device to send INT 10 output to

Return: CX = 0000h output could not be redirected
else INT 10h output now routed to requested display

SeeAlso: AX=6A02h

-----V-106A02-----

INT 10 - DGIS - INQUIRE INT 10 OUTPUT DEVICE

AX = 6A02h
ES:DI = 0000h:0000h

Return: ES:DI = 0000h:0000h if current display is non-DGIS
else address of the current DGIS INT 10 display

SeeAlso: AX=6A00h,AX=6A01h

-----V-106D74-----

INT 10 - MTRFONTS - INSTALLATION CHECK

AX = 6D74h ('mt')
DS:0104h = signature "MtRfOnTs"

Return: AX = 4D54h ('MT') if installed
ES = PSP segment of resident code

Program: MTRFONTS is a shareware TSR (part of the MTR package) by Mechon Mamre
which provides Hebrew screen fonts

SeeAlso: INT 16/AX=6D74h"MTRTSR"

-----V-106E00-----

INT 10 - Paradise VGA internal - GET ???

AX = 6E00h

Return: BX = 5744h ('WD') if supported

DH:AH:AL = last three ASCII digits of ROM serial number

CL = ???

CH = ???

SeeAlso: AX=6E04h,AX=6E05h

-----V-106E04-----

INT 10 - Paradise VGA internal - GET SCREEN SIZE AND ???

AX = 6E04h

Return: BX = screen width (columns)

CX = screen height (lines)

AH = ??? (05h or FFh)

AL = ??? (04h or video mode)

SeeAlso: AX=6E00h,AX=6E05h

-----V-106E05-----

INT 10 - Paradise VGA internal - SET MODE

AX = 6E05h

BL = mode

Return: nothing

Note: like AH=00h, AL=BL.

SeeAlso: AH=00h,AX=6E00h,AX=6E04h,AX=6F05h

-----V-106F00-----

INT 10 - HP Vectra EX-BIOS - INSTALLATION CHECK

AX = 6F00h

BX <> 4850h (usually set to 0000h for simplicity)

Return: BX = 4850h ('HP') indicates HP Extended BIOS video functions present

AX destroyed

SeeAlso: AX=6F00h"Video7",AX=6F01h,AX=6F02h,AX=6F03h,AX=6F04h,AX=6F05h

SeeAlso: INT 14/AX=6F00h,INT 16/AX=6F00h,INT 17/AX=6F00h,INT 33/AX=6F00h

-----V-106F00BX0000-----

INT 10 - VIDEO - Video7 VGA,VEGA VGA - INSTALLATION CHECK

AX = 6F00h

BX = 0000h

Return: BX = 5637h ('V7') indicates Video7 VGA/VEGA VGA extensions are present

SeeAlso: AX=6F01h,AX=6F02h,AX=6F03h,AX=6F04h,AX=6F05h

-----V-106F01-----

INT 10 - VIDEO - Video7 VGA,VEGA VGA,HP EX-BIOS - GET MONITOR INFO

AX = 6F01h

Return: AL = monitor type code (HP,VEGA VGA only) (see #00188)

AH = status register information (see #00189)
CL = current value of Extended Control register (HP Ext BIOS, and only
if AL=41h) (see #00190)

Notes: bits 0-3 are the same as the EGA/VGA status register bits 0-3
supported by original HP Vectra AT and by ES/QS/RS series Vectras

(Table 00188)

Values for monitor type code (HP Extended BIOS):

00h non-HP card with ROM and possibly its own INT 10h driver
41h MultiMode video display adapter
42h-44h reserved
45h industry standard monochrome display adapter
46h industry standard color display adapter
51h reserved

Bitfields for status register information:

Bit(s) Description (Table 00189)

0 display enable
0 = display enabled
1 = vertical or horizontal retrace in progress
1 light pen flip flop set
2 light pen switch activated
3 vertical sync
4 monitor resolution
0 = high resolution (>200 lines)
1 = low resolution (<=200 lines)
5 display type (0 = color, 1 = monochrome)
6,7 diagnostic bits

Note: bits 0-3 are the same as the EGA/VGA status register bits 0-3

-----V-106F02-----

INT 10 - HP Vectra EX-BIOS - SET MONITOR INFO

AX = 6F02h

BL = new value for extended control register (see #00190)

Return: AX,BL destroyed

Notes: this function is only valid when an HP MultiMode Video Display Adapter
is installed

the Extended Control register is at I/O address 3DDh

SeeAlso: AX=6F01h,AX=6F03h

Bitfields for HP Vectra extended control register:

Bit(s) Description (Table 00190)

0 screen resolution (0 = 200 lines, 1 = 400 lines)
 1 underline enable (if set, 'blue' of fg color = underline)
 2 font (0 = Standard-8, 1 = HP-Roman-8)
 3 memory disabled for CPU access
 4 allow access to full 32K memory instead of wrapping at 16K
 5 select second 16K page instead of first
 6,7 unused

-----V-106F03-----

INT 10 - HP Vectra EX-BIOS - MODIFY MONITOR INFO

AX = 6F03h
 BH = exclude mask (set bits are not modified)
 BL = new values for bits indicated by BH (see #00190)

Return: AX destroyed

Notes: this function is only valid when an HP MultiMode Video Display Adapter
 is installed

supported by original HP Vectra AT and by ES/QS/RS series Vectras

SeeAlso: AX=6F01h,AX=6F02h

-----V-106F04-----

INT 10 - VIDEO - Video7 VGA,VEGA VGA,HP Vectra - GET MODE AND SCREEN RESOLUTION

AX = 6F04h
 Return: AL = current video mode (see #00191)
 BX = horizontal columns (text) or pixels (graphics)
 CX = vertical columns (text) or pixels (graphics)

SeeAlso: AX=6F05h

-----V-106F05-----

INT 10 - VIDEO - Video7 VGA,VEGA EXTENDED EGA/VGA,HP Vectra - SET VIDEO MODE

AX = 6F05h
 BL = mode (see #00191)

Return: AX,BL destroyed

Notes: also supported by the HP Vectra Extended BIOS on the original
 HP Vectra AT and by ES/QS/RS series Vectras
 on the HP Vectra, this function rather than AH=00h must be used to
 return to an IBM-standard mode after setting an HP-specified mode
 from 08h to 0Fh.

SeeAlso: AH=00h,AX=0070h,AX=007Eh,AX=6F04h

(Table 00191)

Values for Video7/VEGA video mode:

text/	text	pixel	pixel	colors	disply	scrn	system
grph	resol	box	resolution		pages	addr	
00h-13h	=	standard	IBM	modes	(see	#00010	at AH=00h)

```

08h = T  80x27  .  . mono  . .  HP MultiMode Video
09h = T  80x27  .  .  .  . .  HP MultiMode Video
0Ah = T  40x27  .  . mono  . .  HP MultiMode Video
0Bh = T  40x27  .  .  .  . .  HP MultiMode Video
0Ch = reserved          HP MultiMode Video
0Dh = G  .  .    640x400 .  . .  HP MultiMode Video
0Eh = G  .  .    320x400 .  . .  HP MultiMode Video
0Fh = G  .  .    320x400 .  . .  HP MultiMode Video
40h = T  80x43  8x8  .  .  .  . .  Video7/VEGA VGA
41h = T 132x25  8x14 .  .  .  . .  Video7/VEGA VGA
42h = T 132x43  8x8  .  .  .  . .  Video7/VEGA VGA
43h = T  80x60  8x8  .  .  .  . .  Video7/VEGA VGA
44h = T 100x60  8x8  .  .  .  . .  Video7/VEGA VGA
45h = T 132x28  8x8  .  .  .  . .  Video7/VEGA VGA
60h = G  .  .    752x410 16 . . .  Video7 VGA, VEGA VGA
61h = G  .  .    720x540 16 . . .  Video7 VGA, VEGA VGA
    = G  .  .    720x540 16 . . .  Northgate, Headland 1024i
62h = G  .  .    800x600 16 . . .  Video7 VGA, VEGA Ext EGA
    = G  .  .    800x600 16 . . .  Headland 1024i
63h = G  .  .   1024x768  2 . . .  Video7 VGA
64h = G  .  .   1024x768  4 . . .  Video7 VGA
65h = G  .  .   1024x768 16 . . .  Video7 VGA, VEGA Ext EGA
    = G  .  .   1024x768 16 . . .  Headland 1024i
66h = G  .  .    640x400 256 . . .  Video7 VGA, VEGA Ext VGA
    = G  .  .    640x400 256 . . .  Northgate, Headland 1024i
67h = G  .  .    640x480 256 . . .  Video7 VGA, VEGA Ext VGA
    = G  .  .    640x480 256 . . .  Headland 1024i
68h = G  .  .    720x540 256 . . .  Video7 VGA, VEGA Ext VGA
    = G  .  .    720x540 256 . . .  Headland 1024i
69h = G  .  .    800x600 256 . . .  Video7 VGA, VEGA Ext VGA
    = G  .  .    800x600 256 . . .  Headland 1024i
70h = G  .  .    752x410 16gray . . .  Video7 VGA, VEGA VGA
71h = G  .  .    720x540 16gray . . .  Video7 VGA, VEGA VGA
72h = G  .  .    800x600 16gray . . .  Video7 VGA
73h = G  .  .   1024x768  2gray . . .  Video7 VGA
74h = G  .  .   1024x768  4gray . . .  Video7 VGA
75h = G  .  .   1024x768 16gray . . .  Video7 VGA
76h = G  .  .    640x400 256gray . . .  Video7 VGA
77h = G  .  .    640x480 256gray . . .  Video7 VGA
78h = G  .  .    720x540 256gray . . .  Video7 VGA
79h = G  .  .    800x600 256gray . . .  (future)

```

SeeAlso: AH=00h,#00010,AX=0070h,#00011,AX=007Eh,AX=6F04h,#00083

Index: video modes;Video 7|video modes;VEGA|video modes;Headland

-----V-106F06-----

INT 10 - VIDEO - Video7 VGA,VEGA VGA - SELECT AUTOSWITCH MODE

AX = 6F06h

BL = Autoswitch mode select

00h select EGA/VGA-only modes

01h select Autoswitched VGA/EGA/CGA/MGA modes

02h select 'bootup' CGA/MGA modes

BH = enable/disable (00h enable, 01h = disable selection)

Return: nothing

-----V-106F07-----

INT 10 - VIDEO - Video7 VGA,VEGA VGA - GET VIDEO MEMORY CONFIGURATION

AX = 6F07h

Return: AL = 6Fh

AH = memory configuration

bits 0-6 = number of 256K blocks of video memory

bit 7 = DRAM/VRAM (0: DRAM, 1: VRAM)

BH = chip revision (SR8F) (S/C Chip in VEGA VGA)

BL = chip revision (SR8E) (G/A Chip in VEGA VGA)

CX = 0000h

SeeAlso: AH=12h/BL=10h

-----V-1070-----

INT 10 - VIDEO - TANDY 2000 only - GET ADDRESS OF VIDEO RAM

AH = 70h

Return: AX:BX -> WORD containing green plane's offset

AX:CX -> WORD containing green plane's segment

AX:DX -> WORD containing segment of red (offset 0) and blue (offset
4000) planes

SeeAlso: AH=71h

-----V-1070-----

INT 10 - HUNTER 16 - DRAW ELLIPSE/CIRCLE

AH = 70h

DS:BX -> control block (see #00192)

Return: nothing

Desc: Draws a circle or ellipse. Only valid in graphics modes.

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=71h"HUNTER"

Format of HUNTER 16 control block:

```
Offset Size Description (Table 00192)
00h WORD center X co-ordinate
02h BYTE center Y co-ordinate
03h WORD radius in pixels
05h BYTE color (00h white, 01h black, FFh invert)
06h BYTE aspect ratio x:x (1-127)
07h BYTE aspect ratio y:y (1-127)
```

-----V-107000EX0000-----

INT 10 - Everex Extended Video BIOS - RETURN EMULATION STATUS

AX = 7000h

BX = 0000h

Return: AL = 70h if Trident-based Everex card

CL = monitor type (see #00193)

CH = feature bits (see #00194)

DX = video board info

bits 4-15: board ID model (see #00195)

bits 0-3: board ID revision

DI = BCD BIOS version number

SeeAlso: AX=5F01h,AH=FFh"Oak"

(Table 00193)

Values for Everex monitor type:

```
00h mono
01h CGA
02h EGA
03h digital multifrequency
04h IBM PS/2
05h IBM 8514
06h SuperVGA
07h analog multifrequency
08h super multifrequency
```

Bitfields for Everex feature bits:

Bit(s) Description (Table 00194)

7-6 memory size

00 = 256K

01 = 512K

10 = 1024K

11 = 2048K

5 special oscillator present

4 VGA protect enabled

0 6845 emulation

(Table 00195)

Values for board model for Trident-based Everex cards:

0236h Ultragraphics II
0620h Vision VGA
0673h EVGA
0678h Viewpoint

-----V-107000BX0004-----

INT 10 - Everex Extended Video BIOS - GET PAGING FUNCTION POINTER FOR CURR MODE

AX = 7000h

BX = 0004h

Return: ES:DI -> FAR paging function (call with DL = page to set)

Note: the word preceding ES:DI is the length of the function in bytes, and

the last byte of the function is a FAR return instruction.

SeeAlso: AX=4F05h,AX=7000h/BX=0000h,AX=7000h/BX=0005h

-----V-107000BX0005-----

INT 10 - Everex Extended Video BIOS - GET SUPPORTED MODE INFO

AX = 7000h

BX = 0005h

CL = maximum number of modes to get info for

CH = mode type to get info for (see #00196)

DL = monitor type to get info for

ES:DI -> buffer for mode info (see #00197)

Return: CL = total number of modes fitting criteria

CH = size of each info record

SeeAlso: AX=7000h/BX=0000h,AX=7000h/BX=0004h

(Table 00196)

Values for Everex mode type:

00h all modes
01h monochrome text modes
02h color text modes
03h four-color CGA graphics modes
04h two-color CGA graphics modes
05h 16-color graphics modes
06h 256-color graphics modes

Format of Everex mode information record:

Offset Size Description (Table 00197)

00h BYTE mode number (bit 7 set if extended mode)

01h BYTE mode type (see #00196)
 02h BYTE info bits (see #00198)
 03h BYTE font height
 04h BYTE text columns on screen
 05h BYTE text rows on screen
 06h WORD number of scan lines
 08h BYTE color information
 bits 7-4 reserved
 3-0 bits per pixel

Bitfields for Everex info bits:

Bit(s) Description (Table 00198)

7,6 reserved
 5 monochrome mode
 4 interlaced display
 3 requires special oscillator
 2,1 memory required (00 = 256K, 01 = 512K, 10 = 1024K, 11 = 2048K)
 0 reserved

-----V-1071-----

INT 10 - VIDEO - TANDY 2000 only - GET ADDRESS OF INCRAM

AH = 71h

Return: AX:BX -> WORD containing segment address of INCRAM

AX:CX -> WORD containing offset of INCRAM

SeeAlso: AH=70h"TANDY"

-----V-1071-----

INT 10 - HUNTER 16 - DRAW LINE/BOX

AH = 71h

DS:BX -> control block (see #00199)

Return: nothing

Desc: Draws a line or box. Only valid in graphics modes.

SeeAlso: AH=70h"HUNTER"

Format of HUNTER 16 control block:

Offset Size Description (Table 00199)

00h WORD left edge X co-ordinate
 02h BYTE lower edge Y co-ordinate
 03h WORD right edge X co-ordinate
 05h BYTE upper edge Y co-ordinate
 06h BYTE color (00h white, 01h black, FFh invert)
 07h BYTE type (00h line, 01h box)
 08h BYTE if non-zero (and type is BOX) fill with color

-----V-1072-----

INT 10 - VIDEO - TANDY 2000 only - SCROLL RIGHT PART OR ALL OF SCREEN

AH = 72h

AL = number of columns to shift scroll area, 00h to clear entire area

BH = new attributes for blanked columns at left

CH,CL = row, column of upper left corner of scroll area

DH,DL = row, column of lower right corner of scroll area

Return: nothing

SeeAlso: AH=06h,AH=07h,AH=73h"TANDY",INT 15/AH=12h/BH=05h

-----V-1072-----

INT 10 - HUNTER 16 - SELECT TEXT-MODE INVERSE VIDEO MECHANISM

AH = 72h

AL = mechanism to use

00h do not use inverse video

01h emulate MDA. Display attribute 07h is normal video and 70h is inverse video.

02h use inverse video if background is non-black.

03h use inverse video for high intensity text.

04h use inverse video for text with background intensity greater than foreground intensity.

05h use inverse video for text with background color other than black or high intensity foreground.

Return: nothing

SeeAlso: AH=70h"HUNTER",AH=71h"HUNTER"

-----V-1073-----

INT 10 - VIDEO - TANDY 2000 only - SCROLL LEFT PART OR ALL OF SCREEN

AH = 73h

AL = number of columns to shift scroll area, 00h to clear entire area

BH = new attributes for blanked columns at right

CH,CL = row, column of upper left corner of scroll area

DH,DL = row, column of lower right corner of scroll area

Return: nothing

SeeAlso: AH=06h,AH=07h,AH=72h"TANDY",INT 15/AH=12h/BH=05h

-----V-1073-----

INT 10 - HUNTER 16 - SELECT TEXT FONT

AH = 73h

AL = new font

00h 7x7 in a 8x8 cell

01h 5x7 in a 6x8 cell

06h 3x5 in a 4x6 cell

Return: BH = cell width

BL = cell height

SeeAlso: AH=60h"HUNTER",AH=77h"HUNTER"

-----V-1074-----

INT 10 - HUNTER 16 - SET LCD WINDOWS POSITION

AH = 74h

DH = row (0..24)

DL = column (0..79)

Return: nothing

Desc: set the position in the virtual screen being displayed in the LCD screen. If Split Screen (INT 10/AH=66h) is used, this call sets the position for the window which has the cursor.

SeeAlso: AH=66h"HUNTER",AH=75h, AH=76h

-----V-1075-----

INT 10 - HUNTER 16 - SET ZOOM

AH = 75h

AL = new Zoom state (00h off, nonzero on)

Return: nothing

SeeAlso: AH=74h, AH=76h

-----V-1076-----

INT 10 - HUNTER 16 - LOOKING KEYS

AH = 76h

AL = action

00h home the window

01h move up one line

02h move down one line

03h move left one character

04h move right one character

05h flip window to other side of virtual screen

Return: nothing

Desc: moves the displayed screen around the virtual screen

SeeAlso: AH=74h, AH=75h

-----V-1077-----

INT 10 - HUNTER 16 - GET GRAPHICS FONT

AH = 77h

Return: AL = font number (see #00200)

SeeAlso: AH=60h"HUNTER",AH=73h"HUNTER"

(Table 00200)

Values for HUNTER 16 graphics font number:

00h 7x7 in a 8x8 cell

01h 5x7 in a 6x8 cell

```

02h  7x9  in a  8x10 cell
03h 14x9  in a 16x10 cell
04h  7x18 in a  8x20 cell
05h 14x18 in a 16x20 cell
06h  3x5  in a  4x6  cell

```

Note: size is X * Y

-----V-1078-----

INT 10 - HUNTER 16 - TURN BACKLIGHT ON/OFF

AH = 78h

AL = new state of backlight (00h on, 01h off)

Return: nothing

SeeAlso: AH=64h, INT 15/AH=74h

-----V-107F-----

INT 10 - Paradise SVGA - WD90C24 INSTALLATION CHECK

AH = 7Fh

Return: AX = 1234h if WD90C24 chip installed

-----V-107F00-----

INT 10 - SOLLEX SuperVGA - GET EXTENSIONS INFO

AX = 7F00h

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

ES:DI -> info structure (see #00201)

01h failed

Program: the SOLLEX (Smos videO controLLer EXtensions) SuperVGA functions are an extension to the VESA SuperVGA BIOS Extensions (see AX=4F00h) by Seiko Epson Corporation intended to provided a standardized interface to SuperVGA functionality not addressed by the VESA standard

SeeAlso: AX=4F00h

Format of SOLLEX SuperVGA info structure:

Offset Size Description (Table 00201)

```

00h  DWORD pointer to VESA function dispatch table
04h  DWORD pointer to SOLLEX function dispatch table
08h  DWORD pointer to VESA SuperVGA info (see #00077 at AX=4F00h)
0Ch  DWORD pointer to mode info structure table, consisting of
      alternating ResInfo (see #00202) and VESA mode information
      (see #00079 at AX=4F01h) blocks, terminated with an FFFFh word
10h  DWORD pointer to font info structure table (see #00203)
14h  WORD  high resolution crystal frequency in Hz (0000h = not present)

```

```

16h  DWORD pointer to ASCIZ ID string
1Ah  DWORD pointer to timeout reset table
      array of bytes, each a multiple of the minimum time increment
1Eh  WORD  minimum time increment in timer ticks
20h  BYTE  inverse options supported
      bit 0: inverse supported
      bits 1-7: reserved
21h  BYTE  normal color value
22h  BYTE  inverse color value
23h  WORD  port to be accessed for normal/inverse settings
25h  WORD  type of interface chip (currently undefined)
27h  WORD  program operational mode
      bits 1-0: 00 no preference
              01 terse (minimum detail in program messages)
              10 verbose
              11 use menus if supported, verbose mode otherwise
      bits 7-2: reserved
29h  WORD  SOLLEX specification version
2Bh  WORD  version of VESA/SOLLEX implementation
2Dh  DWORD offset to relocatable portion of SOLLEX extensions (for CONFIG)
2Eh  DWORD offset to unused section of the extensions ROM
31h  16 BYTES reserved

```

Note: all DWORD pointers initially require segment fixups; if the segment is 0000h, it should be changed to the returned ES, otherwise it may be assumed to be correct

Format of SOLLEX SuperVGA ResInfo:

Offset	Size	Description (Table 00202)
00h	WORD	16-bit mode number
02h	WORD	adapter type (00h VGA, 01h EGA, 02h CGA, 03h MDA)
04h	WORD	display info (see #00206)
06h	DWORD	pointer to video parameter table
0Ah	BYTE	replacement entry in master Video Parameter
0Bh	BYTE	mode requested for mode set by BIOS
0Ch	DWORD	pointer to LoadReg table
10h	BYTE	index into table of clock values (see #00204)

SeeAlso: #00201

Format of SOLLEX SuperVGA font info table [array] entry:

Offset	Size	Description (Table 00203)
00h	BYTE	required font height

01h BYTE parameter to load text mode font
02h BYTE parameter to load graphics mode font

SeeAlso: #00201

(Table 00204)

Values for SOLLEX SuperVGA clock value index:

00h 25 MHz
01h 28 MHz
02h PCLK
03h 31.5 MHz (VESA 640x480)
04h reserved
05h 16 MHz (EGA)
06h PCLK
07h 24 MHz (EGA)
08h 25 MHz
09h 28 MHz
0Ah 36 MHz
0Bh 45 MHz (for 1024x768)
0Ch 80 MHz
0Dh 40 MHz
0Eh 65 MHz
0Fh 1 MHz (for powerdown)

SeeAlso: #00201

-----V-107F00BX4000-----

INT 10 - Diamond Stealth 24 - GET S3 INFORMATION BLOCK

AX = 7F00h

BX = 4000h

Return: AX = 007Fh if supported

DX:BX -> DAC set mode routine

SeeAlso: AX=7F00h/BX=4001h,AX=7F00h/BX=4002h,#00732 at INT 1A/AX=B102h

-----V-107F00BX4001-----

INT 10 - Diamond Stealth 24 - GET LINEAR ADDRESS

AX = 7F00h

BX = 4001h

Return: AX = 007Fh if supported

CX = current linear address base (high word)

SeeAlso: AX=7F00h/BX=4000h,AX=7F00h/BX=4002h

-----V-107F00BX4002-----

INT 10 - Diamond Stealth 24 - SET LINEAR ADDRESS

AX = 7F00h

BX = 4002h

CX = new linear address base (high word)

Return: AX = 007Fh if supported

SeeAlso: AX=7F00h/BX=4000h,AX=7F00h/BX=4001h

-----V-107F01BL00-----

INT 10 - SOLLEX SuperVGA - ADAPTER CONTROL - SET ADAPTER

AX = 7F01h

BL = 00h

CX = adapter request (see #00205)

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

Note: initializes video hardware to a particular standard

SeeAlso: AX=7F00h"SOLLEX",AX=7F01h/BL=01h,AX=7F01h/BL=02h

Bitfields for SOLLEX SuperVGA adapter request:

Bit(s) Description (Table 00205)

1-0 adapter type (00 VGA, 01 EGA, 10 CGA, 11 MDA)

2 reserved

4-3 change displays (00 none, 01 analog 10 digital 11 panel)

6-5 desired monitor sense (01 color, 10 mono, 11=8514)

7 lock override

8 alternate adapter mode

15-9 reserved

-----V-107F01BL01-----

INT 10 - SOLLEX SuperVGA - ADAPTER CONTROL - GET ADAPTER

AX = 7F01h

BL = 01h

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

BX = adapter type (see #00205)

DX = display type (see #00206)

SeeAlso: AX=7F00h"SOLLEX",AX=7F01h/BL=00h,AX=7F01h/BL=02h

Bitfields for SOLLEX SuperVGA display type:

Bit(s) Description (Table 00206)

15-9 reserved

```

8 alternate display active
7 Enhanced Color Display monitor active
6 multi-frequency digital monitor active
5 PS/2-type monitor active
4 plasma/electroluminescent panel active
3 LCD panel active
2 multi-frequency analog monitor active
1-0 monitor sense (00 none, 01 color, 10 mono, 11=8514)

```

```
-----V-107F01BL02-----
```

```
INT 10 - SOLLEX SuperVGA - ADAPTER CONTROL - DETERMINE ADAPTER SUPPORT
```

```
AX = 7F01h
```

```
BL = 02h
```

```
CX = adapter request (see #00205)
```

```
Return: AL <> 7Fh if not supported
```

```
AL = 7Fh if supported
```

```
AH = status
```

```
00h requested setting can successfully be made
```

```
01h requested setting not available in this configuration
```

```
SeeAlso: AX=7F00h"SOLLEX",AX=7F01h/BL=00h
```

```
-----V-107F02BL00-----
```

```
INT 10 - SOLLEX SuperVGA - DISPLAY OUTPUT CONTROL - SET DISPLAY OUTPUT
```

```
AX = 7F02h
```

```
BL = 00h
```

```
CX = display output setting (see #00207)
```

```
Return: AL <> 7Fh if not supported
```

```
AL = 7Fh if supported
```

```
AH = status
```

```
00h successful
```

```
01h failed
```

```
SeeAlso: AX=7F00h"SOLLEX",AX=7F02h/BL=00h
```

Bitfields for SOLLEX SuperVGA display output setting:

Bit(s) Description (Table 00207)

```

0 CRTC control in 2,1 valid
1 enable digital output
2 enable analog output
3 panel control in 5,4 valid
4 enable LCD output
5 enable plasma/EL output
6 inverse control in 7 valid
7 0=normal, 1=inverse

```



```
15-8 reserved
-----V-107F02BL01-----
INT 10 - SOLLEX SuperVGA - DISPLAY OUTPUT CONTROL - GET DISPLAY OUTPUT
  AX = 7F02h
  BL = 01h
Return: AL <> 7Fh if not supported
  AL = 7Fh if supported
  AH = status
  00h successful
    BX = display output setting (see #00207)
    CX = displays attached (see #00208)
  01h failed
SeeAlso: AX=7F00h"SOLLEX",AX=7F02h/BL=00h
```

Bitfields for SOLLEX SuperVGA displays attached:

```
Bit(s)  Description (Table 00208)
  0  PS/2 display on analog output
  1  multi-frequency monitor on analog output
  2  LCD panel attached
  3  plasma/electroluminescent panel attached
  4  multi-frequency monitor on digital output
  5  Enhanced Color Display attached to digital outpt
  6  alternate display
15-7 reserved
```

```
-----V-107F03BL00-----
INT 10 - SOLLEX SuperVGA - VIDEO SUPPORT CONTROL - GET SUPPORT INFO
  AX = 7F03h
  BL = 00h
  CX = support type
    0000h VGA, 0001h EGA, 0002h CGA, 0003h MDA, 0004h extensions,
    0005h-0012h reserved for SOLLEX, 0013h Hercules,
    0014h-001Fh reserved for SOLLEX, 0020h-00FFh reserved for OEM
Return: AL <> 7Fh if not supported
  AL = 7Fh if supported
  AH = status
  00h successful
    CX = size of video support code
    DX = segment of physical video support (0000h if no ROM)
    ES = segment of active video support
    ES:DI -> information block (DI = 0000h if none available)
  01h failed
```

SeeAlso: AX=7F03h/BL=01h

-----V-107F03BL01-----

INT 10 - SOLLEX SuperVGA - VIDEO SUPPORT CONTROL - INITIALIZE VIDEO SUPPORT

AX = 7F03h

BL = 01h

CX = support request

ES = segment of support code

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

Note: initializes the indicated video support by calling ES:0003h; this

function may be used to switch the active video support back to

ROM after AX=7F03h/BL=02h

SeeAlso: AX=7F03h/BL=00h

-----V-107F03BL02-----

INT 10 - SOLLEX SuperVGA - VIDEO SUPPORT CONTROL - GO RAM RESIDENT

AX = 7F03h

BL = 02h

CX = support request

ES = destination segment

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

SeeAlso: AX=7F00h"SOLLEX",AX=7F03h/BL=01h

-----V-107F04BL00-----

INT 10 - SOLLEX SuperVGA - POWER CONTROL - SET POWER STATE

AX = 7F04h

BL = 00h

CX = new power state

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

Note: higher values progressively reduce the operations available on the

video adapter while yielding increasing power savings

SeeAlso: AX=7F00h"SOLLEX",AX=7F04h/BL=01h

-----V-107F04BL01-----

INT 10 - SOLLEX SuperVGA - POWER CONTROL - GET POWER STATE

AX = 7F04h

BL = 01h

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

CX = current power state

DX = maximum state

01h failed

SeeAlso: AX=7F00h"SOLLEX",AX=7F04h/BL=00h

-----V-107F04BL02-----

INT 10 - SOLLEX SuperVGA - POWER CONTROL - SET TIMEOUT RESET

AX = 7F04h

BL = 02h

CX = timeout reset

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

SeeAlso: AX=7F00h"SOLLEX",AX=7F04h/BL=03h

-----V-107F04BL03-----

INT 10 - SOLLEX SuperVGA - POWER CONTROL - GET TIMEOUT RESET

AX = 7F04h

BL = 03h

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

BX = time increment

CX = current timeout reset

DX = maximum timeout reset

ES:DI -> timeout reset table (array of bytes)

01h failed

Note: the timeout period is computed as (BYTE ES:[DI+CX]) * BX timer ticks

SeeAlso: AX=7F00h"SOLLEX",AX=7F04h/BL=02h

-----V-107F05-----

INT 10 - SOLLEX SuperVGA - LOAD REGISTER

AX = 7F05h

ES:DI -> register value table (see #00209)

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

SeeAlso: AH=F1h,AH=F3h,AH=F5h

Format of SOLLEX SuperVGA register value table:

Offset Size Description (Table 00209)

00h WORD base I/O register (FFFFh = end of list)

02h 2N BYTES pairs of values to be written to the base I/O register as an
index value and the following register as a data byte

2N+2 WORD FFFFh (end of data list)

... (repeats until FFFFh base address)

-----V-107F06-----

INT 10 - SOLLEX SuperVGA - MULTIPLE FONT CONTROL

AX = 7F06h

BL = subfunction

00h set multiple font state

CX = new state (00h off, 01h on)

01h get multiple font state

Return: BL = current state (00h off, 01h on)

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

SeeAlso: AX=1100h,AX=1103h

-----V-107F07-----

INT 10 - SOLLEX SuperVGA - FILL VIDEO RAM

AX = 7F07h

BL = how much to fill

00h regen size

01h all video memory

CX = pattern to write (normally 0720h for text modes and 0000h for gr)

Return: AL <> 7Fh if not supported

AL = 7Fh if supported

AH = status

00h successful

01h failed

SeeAlso: AH=06h

-----A-1080-----

INT 10 - CU Writer v1.4 - GET OPTION

AH = 80h

AL = desired option

Return: ???

Program: CU Writer is a public-domain Thai-English word processor from
Chulalongkorn University, Bangkok

SeeAlso: AH=81h"CU Writer"

-----Q-1080--DX4456-----

INT 10 U - VIDEO - DESQview 2.0x only - internal - SET ??? HANDLER

AH = 80h

DX = 4456h ('DV')

ES:DI -> FAR subroutine to be called on ???

Return: DS = segment of DESQview data structure for video buffer

Note: this function is probably meant for internal use only, due to the magic
value required in DX

the subroutine seems to be called when the DESQview menu is accessed;
on entry, AL = 03h or 04h

-----A-1081-----

INT 10 - CU Writer v1.4 - SET OPTION

AH = 81h

???

Return: ???

SeeAlso: AH=80h"CU Writer"

-----Q-1081--DX4456-----

INT 10 U - VIDEO - DESQview 2.0x only - internal - GET ???

AH = 81h

DX = 4456h ('DV')

Return: ES = segment of DESQview data structure for video buffer

BYTE ES:[0] = current window number in DV 2.0x

Note: this function is probably meant for internal use only, due to the magic
value required in DX

SeeAlso: AH=82h"DESQview"

-----Q-1082--DX4456-----

INT 10 U - VIDEO - DESQview 2.0x only - internal - GET CURRENT WINDOW INFO

AH = 82h

DX = 4456h ('DV')

Return: DS = segment in DESQview for data structure

in DV 2.00,

BYTE DS:[0] = window number

WORD DS:[1] = segment of other data structure
WORD DS:[3] = segment of window's object handle
ES = segment of DESQview data structure for video buffer
AL = current window number
AH = ???
BL = direct screen writes
 00h program does not do direct writes
 01h program does direct writes, so shadow buffer not usable
BH = ???
CL = current video mode
CH = ???

Note: this function is probably meant for internal use only, due to the magic value required in DX

SeeAlso: AH=81h"DESQview"

-----J-108200-----

INT 10 - VIDEO - AX PC - GET/SET SCROLL MODE

AX = 8200h

BL = new scroll mode or FFh to get current mode

 00h dynamic, 01h software

Return: AL = scroll mode (current mode if BL=FFh, previous mode otherwise)

SeeAlso: AH=06h,AH=07h

-----J-108300-----

INT 10 - VIDEO - AX PC - GET VIDEO RAM ADDRESS

AX = 8300h

Return: AX = offset of video RAM

ES:BX -> virtual text RAM buffer

SeeAlso: AX=5201h

-----1086-----

INT 10 - ???

AH = 86h

Note: called by Diamond Stealth64 Video STLTH64.VXD

SeeAlso: INT 2F/AX=4021h

-----N-108B-----

INT 10 - Alloy MW386 - FORCE WORKSTATION SCREEN UPDATE

AH = 8Bh

Return: nothing

SeeAlso: AH=92h,AH=93h

-----N-1090-----

INT 10 - Alloy MW386 - GET PHYSICAL WORKSTATION DISPLAY MODE

AH = 90h

Return: AL = current video mode (see #00010 at AH=00h)

SeeAlso: AH=91h

-----N-1091-----

INT 10 - Alloy MW386 - GET PHYSICAL WORKSTATION ADAPTER TYPE

AH = 91h

Return: AL = video adapter type (see #00210)

Note: types less than 80h do not imply that the current user is on the host

SeeAlso: AH=90h

(Table 00210)

Values for Alloy MW386 video adapter type:

00h monochrome

01h Hercules monochrome graphics

02h CGA

03h EGA

04h VGA

80h monochrome text terminal

81h Hercules graphics terminal

82h color graphics terminal

-----N-1092-----

INT 10 - Alloy MW386 - INHIBIT WORKSTATION SCREEN UPDATES

AH = 92h

Return: nothing

Note: the terminal will be updated even when screen updates are inhibited if

TTY output is used

SeeAlso: AH=8Bh

-----N-1093-----

INT 10 - Alloy MW386 - REDRAW SCREEN

AH = 93h

Return: nothing

SeeAlso: AH=8Bh,AH=FFh

-----V-109508-----

INT 10 - VHRBIOS.SYS - ???

AX = 9508h

Return: ???

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

black&white A4/portrait monitor

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

-----V-1096-----

INT 10 - VHRBIOS.SYS - GET OTHER DISPLAY INFO

AH = 96h

Return: AL = display type (see AH=1Ah)

AH = ??? (possibly related to Micro Channel support)
Program: VHRBIOS.SYS is a driver for the Micro Display Systems "TheGenius"
 black&white A4/portrait monitor
Desc: returns info on any non-Genius video adapter in the system
SeeAlso: AX=9508h"VHRBIOS.SYS",AH=F0h"VHRBIOS.SYS"
-----A-10A0-----
INT 10 - CU Writer v1.4 - SET PIXEL WRITE MODE
 AH = A0h
 ???

Return: ???
Program: CU Writer is a public-domain Thai-English word processor from
 Chulalongkorn University, Bangkok
SeeAlso: AH=80h"CU Writer",AH=B0h
-----p-10A000-----
INT 10 - ATI BIOS Extensions - ???
 AX = A000h
 CL = ??? (bits 2-0 may not be 000)
 CH = ??? (81h or ???)
 DX = segment of ??? (if CH=81h)
 ???

Return: AX = ???
Program: the ATI BIOS extensions are supported by ATI graphics cards containing
 the Mach64 or newer graphics chips (Rage Pro, Rage 128, etc.)
SeeAlso: AX=4F10h/BL=00h,AX=A001h
-----10A001-----
INT 10 - ATI Mach64 BIOS - ???
 AX = A001h
 CL = ??? flags
 ???

Return: AH = status
 00h successful
 01h failed
 ???

SeeAlso: AX=A000h,AX=A002h
-----10A002-----
INT 10 - ATI Mach64 BIOS - ???
 AX = A002h
 CL = ??? (bits 2-0 may not be 000)
 CH = ??? (81h or ???)
 DX = segment of ??? (if CH=81h)
 ???

Return: AH = status
00h successful
01h failed
???

Note: executes AX=A000h, then AX=A001h if A000h was successful

SeeAlso: AX=A000h,AX=A001h

-----10A003-----

INT 10 - ATI Mach64 BIOS - ???

AX = A003h

BX = ???

???

Return: AH = status (00h successful)

???

SeeAlso: AX=A000h,AX=A002h

-----V-10A004-----

INT 10 - PhoenixVIEW/LC - VIDEO DISPLAY TYPE

AX = A004h

???

Return: ???

Desc: configure display parameters

SeeAlso: AX=A01Bh"PhoenixVIEW/LC"

-----10A004-----

INT 10 - ATI Mach64 BIOS - ???

AX = A004h

BX = video mode??? (0000-00FFh)

DX = ??? (sent to some port [66ECh???) one bit at a time, high to low)

???

Return: AH = status

00h successful

01h failed

???

SeeAlso: AX=A000h,AX=A002h

-----10A005-----

INT 10 - ATI Mach64 BIOS - ???

AX = A005h

CL = ??? flags (bits 0,2, others?)

???

Return: AH = status

00h successful

01h failed

???

SeeAlso: AX=A000h,AX=A002h

-----10A006-----

INT 10 - ATI Mach64 BIOS - ???

AX = A006h

???

Return: AH = status

00h successful

01h failed

???

SeeAlso: AX=A000h,AX=A002h

-----10A007-----

INT 10 - ATI Mach64 BIOS - GET ???

AX = A007h

Return: AH = 00h (successful)

AL = ??? (00h)

BX = ??? (15A0h in examined BIOS)

DX = segment of BIOS

SeeAlso: AX=A000h,AX=A002h

-----10A008-----

INT 10 - ATI Mach64 BIOS - GET ??? SIZE

AX = A008h

CL = ??? flags

???

Return: AH = 00h (successful)

BX = size of ??? (0046h if CL bit 0 clear on entry)

SeeAlso: AX=A000h,AX=A009h

-----10A009-----

INT 10 - ATI Mach64 BIOS - GET ???

AX = A009h

DX:BX -> buffer for ???

CL = ??? flags

???

Return: AH = status

00h successful

DX:BX buffer filled

01h failed

???

SeeAlso: AX=A000h,AX=A008h

-----10A00A-----

INT 10 - ATI Mach64 BIOS - GET ???

AX = A00Ah

Return: AH = 00h (successful)

AL = ??? (01h)

BX = ??? (5938h)

CX = ??? (595Eh)

DX = segment of BIOS

SeeAlso: AX=A000h,AX=A002h

-----10A00B-----

INT 10 - ATI Mach64 BIOS - ???

AX = A00Bh

BX = ???

CL = ??? flags

???

Return: AH = status

00h successful

01h failed

???

SeeAlso: AX=A000h,AX=A002h

-----p-10A00C-----

INT 10 - ATI Mach64 BIOS - SET SLEEP MODE

AX = A00Ch

CL = new state

00h normal operation

01h standby

02h suspend

03h off

Return: AH = status

00h successful

01h failed

Note: CL bit 2 affects values written to video card's ports

SeeAlso: AX=A000h,AX=A00Dh,AX=4F00h,AX=4F10h/BL=01h

-----10A00D-----

INT 10 - ATI Mach64 BIOS - GET SLEEP MODE

AX = A00Dh

Return: AH = 00h (successful)

CL = current power mode

00h normal operation

01h standby

02h suspend

03h off

SeeAlso: AX=A00Ch,AX=4F10h/BL=02h

SeeAlso: AX=A00Ch,AX=A00Dh"Rage128"AX=4F10h/BL=02h

-----V-10A00D-----

INT 10 - ATI Rage128 BIOS - GET ???

AX = A00Dh

BL = subfunction

00h ???

01h ???

02h ???

03h ???

DX:DI -> buffer for results

???

Return: AH = status

00h successful

01h failed

???

SeeAlso: AX=A000h,AX=A00Dh"Mach64",AX=A015h

-----10A00E-----

INT 10 - ATI Mach64 BIOS - SET ???

AX = A00Eh

CL = ??? (00h-03h)

Return: AH = status

00h successful

01h failed

Notes: CL zero/nonzero are treated differently

NOP in All-in-Wonder 128 BIOS

SeeAlso: AX=A000h,AX=A00Fh

-----10A00F-----

INT 10 - ATI Mach64 BIOS - GET ???

AX = A00Fh

Return: AH = 00h (successful)

CL = ??? (00h-03h, read from PORT 42ECh)

Note: NOP in All-in-Wonder 128 BIOS

SeeAlso: AX=A000h,AX=A00Eh

-----10A010-----

INT 10 - ATI Mach64 BIOS - ???

AX = A010h

CL = ??? (80h/other)

???

Return: AH = status

00h successful

01h failed

???

Note: NOP in All-in-Wonder 128 BIOS

SeeAlso: AX=A000h,AX=A002h

-----10A011-----

INT 10 - ATI Mach64 BIOS - GET ???

AX = A011h

Return: AH = 00h (successful)

BL = ??? (1Fh)

BH = ??? (0Fh)

CL = ??? (00h)

CH = ??? (0Fh)

DX = ??? (00FFh)

Note: NOP in All-in-Wonder 128 BIOS

SeeAlso: AX=A000h,AX=A010h

-----V-10A012-----

INT 10 - ATI Mach64 BIOS - GET ???

AX = A012h

Return: AH = 00h (successful)

AL = ??? (00h)

BX = ??? (0000h)

CX = ??? (0000h)

DX = ??? (02ECh)

Note: NOP in All-in-Wonder 128 BIOS

SeeAlso: AX=A000h,AX=A002h

-----V-10A013-----

INT 10 - ATI Mach64 BIOS - ???

AX = A013h

???

Return: AH = status

00h successful

01h failed

???

Note: not implemented in examined Mach64 BIOS, always returns AH=01h; NOP in
examined All-in-Wonder 128 BIOS

SeeAlso: AX=A000h,AX=A002h

-----V-10A014-----

INT 10 - ATI Rage128 BIOS - ???

AX = A014h

CH = ??? (bits 0,1,4)

???

Return: AH = status

00h successful

```
    01h failed
    ???
SeeAlso: AX=A000h,AX=A015h
-----V-10A015-----
INT 10 - ATI Rage128 BIOS - GET/SET ???
    AX = A015h
    CH = subfunction
    00h get ???
    01h set ???
        CL = ??? (only bits 0,1 used on All-in-Wonder 128)
    other NOP
Return: AH = status
    00h successful
    01h failed
    ???
SeeAlso: AX=A014h,AX=A017h
-----V-10A016-----
INT 10 - ATI Rage128 BIOS - NOP
    AX = A016h
Return: nothing
SeeAlso: AX=A000h,AX=A017h
-----V-10A017-----
INT 10 - ATI Rage128 BIOS - ???
    AX = A017h
    ???
Return: AH = status
    00h successful
    01h failed
    ???
SeeAlso: AX=A014h,AX=A015h
-----V-10A01B-----
INT 10 - PhoenixVIEW/LC - CRT MONITOR DETECTION
    AX = A01Bh
    ???
Return: ???
Desc: determine whether an external display is currently attached
SeeAlso: AX=A004h"PhoenixVIEW/LC"
-----U-10AA-----
INT 10 OU - Player's Tool 3.0 - UNINSTALL
    AH = AAh
Return: ???
```

Note: this call has been removed in PT 3.996b

Program: Player's Tool is a game cheater by Dmitry Yakunin & Andy Robinson

SeeAlso: AH=ABh,INT 2F/AH=AAh

-----U-10AB-----

INT 10 OU - Player's Tool 3.0 - INSTALLATION CHECK

AH = ABh

Return: BH = FFh

???

Note: this call has been removed in PT 3.996b

SeeAlso: AH=AAh,INT 2F/AH=ABh

-----A-10B0-----

INT 10 - CU Writer v1.4 - PUT PIXEL

AH = B0h

???

Return: ???

SeeAlso: AH=A0h"CU Writer",AH=B1h,AH=B2h,AH=B3h,AH=B4h,AH=B5h,AH=D0h

-----A-10B1-----

INT 10 - CU Writer v1.4 - MOVE TO

AH = B1h

???

Return: ???

SeeAlso: AH=B0h,AH=B2h,AH=C0h"CU Writer"

-----A-10B2-----

INT 10 - CU Writer v1.4 - DRAW LINE TO

AH = B2h

???

Return: ???

SeeAlso: AH=B0h,AH=B1h,AH=B3h,AH=B4h,AH=B5h

-----A-10B3-----

INT 10 - CU Writer v1.4 - FLOOD FILL

AH = B3h

???

Return: ???

SeeAlso: AH=B2h,AH=B5h

-----A-10B4-----

INT 10 - CU Writer v1.4 - DRAW RECTANGLE

AH = B4h

???

Return: ???

SeeAlso: AH=B0h,AH=B2h,AH=B3h,AH=B5h

-----A-10B5-----

INT 10 - CU Writer v1.4 - DRAW FILLED RECTANGLE

AH = B5h

???

Return: ???

SeeAlso: AH=B0h,AH=B2h,AH=B3h,AH=B4h

-----V-10BB-----

INT 10 U - Doorway - SET BIOS/DIRECT REDIRECTION

AH = BBh

AL = new redirection mode

00h BIOS (hook and redirect INT 10h,etc.)

01h DIRECT (scan video memory for changes)

Return: ???

SeeAlso: INT 16/AH=67h"Doorway"

-----V-10BD44-----

INT 10 U - Compaq QVision - ???

AX = BD44h

???

Return: ???

Note: called by Compaq's MTX132.COM, which places the QVision in 132-column
mode

-----!---Section-----