486 Monitor's System Call Interface

You can call up the Monitor utility programs by entering the system call: "INT 20 (dec)" with a Function Code in register <BL>; and if required a Parameter Block Pointer in <DS:AX>. (See Table 6-4).

NOTE

"input" - Refers to the parameters required by the PROMs. You must supply it before typing the INT 20 request.

"output"- Is the result returned to you from the PROM.

The following Monitor system calls are supported. See Table 6-4.

Table 6-4 Monitor System Calls

Function Code (dec)	System Call	Function Description

01	con_in	Read in an ascii character from the keyboard console
		
		(Note: This routine waits until a character is typed on the keyboard)
		
		[input: bl-01; int 20]
		
		[dx = 0 = Z80'S channel 0;
		dx = 1 = Z80's channel 1;
		dx = 2 = Z80'S channel 2;
		dx = 3 = Z80'S channel 3;
		dx = Off = Diag Console (8274)]
		
		[output: al=ascii character]

02	con_out	Write one character to the console [input: al=ascii character; bl-02; dx has the same assignment as con_in, int 20] [output: none]
03	con_str	Write a character string to the console [input: ds:ax = character array pointer ending with a "0"] [bl-03; int 20; dx has the same assignment as con_in] [output: none]

Table 6-4 Monitor System Calls

Function Code (dec)	System Call	Function Description
11	con_stat	Read in the keyboard status [input: bl=11; int 20 ;dx has the same assignment as in con_in] [output: al = 0 means no character; al = 1 means caracter available] (Note: The "con-stat" and "con-in" calls should be prepared together. You should first call "con-stat" to check if there is a char-acter available then call a "con-in" to read this character. See "con-in" above.
12	auto_reboot	Reboots the system from the device requested and also restarts the monitor operation, while skipping over the power-up tests [input: bl=12; int 20 ;bh = device number] [bh = 0 = default auto_boot] [output: none]
13	boot_number	Returns the boot-device number [refer to the power-up test descriptions for device assignment] [input: bl=13; int 20]

Table 6-4 Monitor System Calls

Function Code (dec)	System Call	Function Description
30	general I/O	Performs the peripheral (including Floppy, Hard Disk, Network, tape, etc.) operation specified by the I/O parameter block defined in the following protocol. [input: bl=30; int 20; ds:ax=pointer to parameter block] [output: resultant status return in the pre- defined locations of the same parameter block]

Table 6-5 Peripheral I/O Parameter Block Layout

Relative Byte Address	Byte Content
000(H)	[Device Number (low byte)]
001(H)	[Device Number (high byte)]
002(H)	[Command (low byte)]
003(H)	[Command (high byte)]
004(H)	[Result (low byte)]
005(H)	[Result (high byte)]
006(H)	[Device Result (Status byte 1)]
007(H)	[Device Result (Status byte 2)]
008(H)	[Device Result (Status byte 3)]
009(H)	[Device Result (Status byte 4)]
00A(H)	[DMA Segment (low byte)]
00B(H)	[DMA Segment (high byte)]
00C(H)	[DMA Offset (low byte)]
00D(H)	[DMA Offset (high byte)]
00E(H)	[Cylinder (low byte)]
00F(H)	[Cylinder (high byte)]
010(H)	[Drive Number]
011(H)	[Head Number]
012(H)	[Beginning Sector Number]
013(H)	[Sector Count Number]
014(H)	[Sector length]
015(H)	[Retries]
016(H)	

Total Block Size = 22 (dec) bytes

Refer to Disk I/O Parameter Block Definition

Table 6-6 Peripheral I/O Parameter Block Definition

	Disk Only	
No. Device	No. Command	No. Status
0 - Hard Disk	0 - Recal	0 - No Error
1 - Floppy Disk	1 - Write Sector	1 - General Error
2 - SCSI (Not Used)	2 - Read Sector	2 - Device not supported
3 - Tape	4 - Seek	3 - Device not present
4 - Printer	5 - Format	4 - Invalid command
5 - WorkNet		5 - Timeout error
6 - EtherNet(Not Used)		
7 - Console		
8 - Auto Boot		
9 - 586 Floppy I/O (Not Used)		

Res_1 to Res_4 will have the device's results.

(Refer to individual device driver's specification. However, a non-zero normally indicated a failure condition)

The beginning sector byte is used to specify the gap length size during hard disk and floppy formats. (Note: typically, a value of 50 (hex) is used. In both cases, the format requires the DMA buffer to be filled according to the format command specifications command required by the controller chip. (Note: Refer to individual device driver's specification).