MAKING BIOS CALLS
(continued)

DMS BIOS CALLS

DMS has implemented interrupt 78 as a COMMON BIOS entry point. The user stuffs the registers required by the specific BIOS call, moves the COMMON BIOS function number to the AH register and then issues an INT 78. THIS METHOD MAY ALSO BE USED BY CP/M-86 PROGRAMS, HOWEVER THE 'COMMON' FUNCTION NUMBER (NOT THE CP/M-86 FUNCTION NUMBER) MUST BE USED.

The SI, BP and DS registers are saved. **All others (including the flags) may be modified by the bios.** The stack is manipulated to insure the flags, as modified by the BIOS, are returned to the user.

CP/M-86 and MS-DOS

To keep a semblance of compatibility the parameter block passed by the application is the same for CP/M-86 and MS-DOS. It is 5 bytes in length and contains:

BYTE	
0	BIOS function Number
1-2	value to be loaded into CX
3-4	value to be loaded into DX

If the function requires more data than can be passed in CX and DX DX:CX will be the segment:offset of a parameter block tailored for the function being called. For example - the parameter block used by SNDNET and RCVNET is:

BYTE	
0	User number
1-2	offset of data to send/receive
3-4	segment of data to send/receive
5-6	length of data to send/receive

The next two pages show the currently implemented BIOS calls available in all three operating systems. The Z80 and the Z816 share the same function numbers, however they may not always expect the same parameters or work exactly the same way. See the section on the particular function desired for specifics. CP/M-86 uses the function numbers in the CPM-86 column when making BIOS calls via the BDOS (function 50). If using INT 78 the function numbers in the COMMON column should be used. MS-DOS uses the function numbers from the COMMON column for both the DMSBIOS$ and the INT 78 methods of accessing the BIOS. THE INT 78 METHOD IS PREFERRED.

BIOS Function Numbers

BIOS Function	CPM-80 function #	CPM-86 (bdos 50) function #	COMMON (int 78) function #
Cold Boot	-1	-1	-
Warm Boot	0	0	-
Console Status	1	1	0
Console Flush	-	-	1
Console Input	2	2	2
Console Output	3	3	3
List Output	4	4	5
Punch Output	5	5	-
Reader Input	6	6	-
Aux Status	-	-	6
Aux Flush	-	-	7
Aux Input	-	-	8
Aux Output	-	-	9
HOME disk	7	7	-
Select disk	8	8	-
Set Track	9	9	-
Set Record	10	10	-
Set DMA offset	11	11	-
Read Record	12	12	10
Write Record	13	13	11
List Status	14	14	4
Get Memory size	-	-	12
Media Same	-	-	13
Spool Flush	-	-	14
Set I/O pointer	-	-	15
Clear Locks	-	-	16
Aux Status 1	-	-	17
Record Translation	15	15	-
Set DMA segment	-	16	-
Get Segment Table	-	17	-
Get I/O byte	-	18	19
SET I/O byte	-	19	20

BIOS Function Numbers
(continued)

BIOS Function	CPM-80 function #	CPM-86 (bdos 50) function #	COMMON (int 78) function #
Make Assignment	-	20	21
Get Assignment	-	21	22
Set/reset visible error flag	-	24	24
Get Error Count	-	25	25
Set/reset Retry flag	-	26	26
WHO request	-	27	27
DIRNET request	-	28	28
Flush Flop Buff	30	-	-
Net Lock	31	29	29
CP/M MAP	32	-	-
Net Unlock	33	30	30
Set I/O byte count	34	-	-
Version (location of version)	35	-	-
SENDNET/sndnet	36	37	37
RECNET/rcvnet	37	38	38
NACKPOLL/clrpol	38	36	36
ACKPOLL/setpol	39	35	35
'USER" port output	40	-	-
Set poll prime address	41	32	32
Set Receive timeout	42	-	-
Local Hard Disk Status	43	-	-
Net Hard Disk status	44	33	33
Set Network Mode	45	31	31
Set List Type	46	-	-
Hard Disk Reset	47	-	-
Write Mode Request	48	-	42
BIOS information	49	-	41
Time and Date	-	34	34
Direct Gut bios call	-	39	-
816 Video Routine	-	-	39
Print Screen	-	-	40
Aux Init	-	-	43
Partition Table Address	-	-	44
Network Info Request	-	-	45
Set Print Escape	-	-	46
Spool Message flip/flop	-	-	47

GUT (common) BIOS CALLS FOR 8086/88 DEVICES

This document describes the calls the can be made to the GUTBIOS on DMS 8086 and 8088 based machines. The SYSTEM bios forCPM80 (on 816s),CPM86 and MSDOS all use some of these calls. They can be called by a user through two methods:

A. DIRECT BIOS CALL through BDOS.

BDOS call 50 calls the BIOS which can then call the GUTBIOS. To make the call you must set up two buffers: one for the BDOS to use to call the BIOS, and one for the BIOS to use to call the GUTBIOS.

Here is an example:

To call GUTBIOS function 33 HDSTAT.

HDSTAT expects DX and CX to hold the address of the buffer to read information into. Say the buffer is located at address DATASEG:HDBUFFER.

Set up a buffer for call to HDSTAT from the BIOS called BIOScall (in DATASEG segment). This buffer contains

DATASEQ equ 1000h (segment this program is running in)

```
1 byte   - GUTBIOS FUNCTION NUMBER
                (in this case 33  (HDSTAT Call))
1 word   - VALUE to load into CX for GUTBIOS call
                (in this case offset of HDBUFFER)
1 word   - VALUE to load into DX for GUTBIOS call
                (in this case value of DATASEG)
```

In code:

```
BIOScall        db      33
                dw      HDBUFFER
                dw      DATASEG
```

Now set up a buffer (BDOScall) for BDOS to call BIOS with

```
1 byte   - BIOS FUNCTION NUMBER
                (always 40 (28h) - for Call to gut)
1 word   - VALUE to load into CX for BIOS call
                (in this case offset of BIOScall)
1 word   - VALUE to load into DX for GUTBIOS call
                (in this case value of DATASEG)
```

In code:

```
                db      28h
                dw      BIOScall
                dw      DATASEG
```

GUT (common) BIOS CALLS FOR 8086/88 DEVICES
(continued)

Now to call the gut function we do this:
(assuming DS already has value DATASEG in it)

```
mov     dx,offset BDOScall      ; buffer with call to bios
mov     cx,50                   ; bios call function
int     0E0h                    ; call bdos
```

B. USING INTERRUPTS TO CALL GUT BIOS FUNCTIONS:

For all functions except video functions:

Put function number in AH.

Load all registers as Function you are calling expcects them.

call interupt 78.

With the example above:

```
mov ah,33
mov cx,offset HDBUFFER
mov dx,ds
int 78
```

816 Video functions will be called directly thorugh interrupt 79 when it is implemented, possibly release 7.

GUT (common) BIOS CALLS FOR 8086/88 DEVICES
(continued)

INDEX OF GUTBIOS FUNCTION NUMBERS

conStatus	0
conFlush	1
auxFlush	7
auxInit	43
auxInput	8
auxOutput	9
auxStatus	6
auxStat1	17
ClrLock	16
clrpol	36
COERCE	31
conInput	2
conOutput	3
DIRNET	28
diskRead	10
diskWrite	11
DMSinfo	41
GETASS	22
GETERR	25
GETIOBF	19
getMemSize	12
HDSTAT	33
LOCK	29
MAKASS	21
mediaSame	13
netInfoReq	45
partAddr	44
printScreen	40
prnOutput	5
prnStatus	4
RCVNET	38
SETERR	24
SETIOBF	20
setIOptr	15
setpol	35
SETPPA	32
setPResc	46
SETTRY	26
SNDNET	37
spoolFlush	14
spoolMess	47
TIMDAT	34
UNLOCK	30
VIDEO	39
WHO	27
WRScommand	42

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)

0. CONSTATUS - get status of console input.

ON ENTRY: nothing
ON EXIT: If there is a char the zero flag is reset and char is in AX. Otherwise zero flag is set, 0 in AL.

1. CONFLUSH - empty the keyboard type ahead buffer and input port. This routine keeps getting chars and throwing them away till there ain't no more.

2. CONINPUT - Get character from console.

ON EXIT: AX hold char. Routine does not return until it has a char.

3. CONOUTPUT - Outputs a character to the console.

ON ENTRY: Char is in AL.
ON EXIT: nothing.

4. PRNSTATUS - check printer status.

ON EXIT:
For all devices except spooler:
Zero flag reset if printer is ready to fly.
For Spooler: If a falied attempt was made to print a char since the last job end, the status is bad, and every time status is called another attempt is made to open a spool block. Status is based on the results of this attempt.

If ready: zero flag reset, al has FF
If not ready: zro flag set, al has 0

5. PRNOUTPUT - output a char to the list device.

ON ENTRY: char is in AL.
ON EXIT:
IF SPOOLER
char is in AL
success - zero flag reset.
failure - zero flag set.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

6. AUXSTATUS - get status from the auxilliary output device.

   ON EXIT: AL contains status from Read Reg 0 of SIO Zero flag is reset if device is ready to transmit.

7. AUXFLUSH - empty the auxilliary buffer, if any.
   NOT IMPLEMENTED.

8. AUXINPUT, - get char from auxilliary device.

   ON EXIT: char is in AL.

9. AUXOUTPUT - write char to AUX device

   ON ENTRY: char is in AL.
   ON EXIT: nothing

10. DISKREAD - read from disk.

    ON ENTRY: al - logical drive number
              es - buffer segment
              di - buffer offset
              dx - sector
              bx - track
              cx - number of 128b sectors to read
    ON EXIT: Info is read into the buffer at es:di. If success zero flag is reset Else zero is set and cl has number of sectors successfully read.

11. DISKWRITE - parameters identical to disk read. (see above)

12. GETMEMSIZE - returns the physical size of memory.

    ON EXIT: cx has memory size in (16 byte) paragraphs.

13. MEDIASAME - return the current value of the media change flag, and reset the media chagne flag to true. Used by MSDOS only, to see if it needs to re-read a directory before doing some file IO. Media change flag is set true whenever there is an assign to make MSDOS keep up with the latest.

14. SPOOLFLUSH - Ends a spool job if one is active.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

15. SETIOPTR - relocates the iobyte to the address asked for.

ON ENTRY: cx has offset of address.
dx has segment of address.
If addr is 0:0 it means reset to default IOBYTE address.
ON EXIT: moves the current IOBYTE to this address, and thereafter accesses that address when it needs IOBYTE.

16. CLRLOCK - sends byte to master that says clear all locks for our user number.

ON ENTRY: nothing
ON EXIT: AX is 0, zero flag set.

17. AUXSTATUS1 - Get status of aux device from SIO read reg 1. This call is needed only by the rombios.

19. GETIOBF - Get the iobyte.

ON EXIT: al has the IOBYTE.

20. SETIOBF - Set the iobyte.

ON ENTRY: value to set IOBYTE is in cl.

21. MAKASS - This poor bastard attempts to assign a partition to a logical drive number. The only valid drive types are HiNet and memory disk. (IMPORTANT: Memory disk is only valid on CPM 86 since MSDOS does not allow bios manipulation of the memory allocation table. An attempt to assign a memory disk while running under MSDOS will result in a crashola.)

The format of the assign request buffer is:

byte 0	-	logical drive number	(set by caller)
1-8	-	partition name	(set by caller)
9	-	partition size	
10	-	partition number	
11	-	control byte	(caller sets OS type)
12	-	volume number	
13	-	drive type	(set by caller)
14	-	write status	
15-20	-	password	(set by caller)

OS type set by caller in control byte: Low bit determines OS

CPM80,HIDOS, & CPM86 - 0
MSDOS - 1

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

The possible drive type values are listed here. Only the first two are currently valid for the 86 station. The rest are set found in use with Z80 stations:

Network Partition	-	60h
Memory Disk	-	E0h
8 inch Hard Disk	-	40h
5 inch Hard Disk	-	C0h
mini-floppy(one side)	-	80h
mini-floppy(one side)	-	A0h
8" floppy (single dn)	-	20h
8" floppy (double dn)	-	00h

ON ENTRY: cx has offset of request buffer
dx has segment of request buffer

ON EXIT:
IF SUCCESS
AL = 0 and all bytes in buffer not set by caller are filled in by bios. Note that the write status field is set on the basis of the control byte, so if its an ownable partition it is marked as unowned. No query about its present status is made by this module.

IF FAIL
AL = 1 assignment invalid.
AL = 2 invalid partition name or password
AL = 3 insufficient memory for memdisk.
AL = 4 mem disk already in use.
AL = 6 partition is not of requested OS type.

NOTE: even if an assign fails the ownership of the previously assigned partition on that logical drive is released.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

22. GETASS: Get information about the parttion currenlty assigned to a logical drive.

Format of buffer:

byte 0	-	logical drive number (set by caller)
1-8	-	partition name
9	-	partition size
10	-	partition number
11	-	control byte
12	-	volume number
13	-	drive type
14	-	write status
15	-	unused
16	-	unusud

ON ENTRY: DX:CX is addr of buffer.

ON EXIT:
IF SUCCESS
Buffer is filled out with all current info.
AL = 0

IF FAILURE
AL = 5 meaning drive number in request was illegal.

NOTE: A partition number of 0FFh means drive is unassigned.

ON PARTITON WRITE OWNERSHIP:
Whenever a network partition is assigned, the control byte is checked. If the c.b. indicates a write status of R/W, R/O or HiDos the write status is set to that value. Otherwise the write status is set to unknown. Whenever a memdisk is assigned its write status byte is set to read/write.

On OS initialization if any of user's default partition names have the high bit set on the first character, the OS attempts to gain ownership status for that drive through a network ownership request. If ownership is denied a message is given to user warning them that they cannot write to that drive.

Whenever a write is attempted, the write status of the drive is checked. If it is not equal to the user number or read/write status or shared (when running HIDOS) the user is informed that they can't write to that partion, and a failure is returned. CPM then warm boots while MSDOS will query what user wants. No matter what they say, they cannot write to the partition till they abort and do an explicit write onwership request through ASSIGN.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

Whenever an owned drive is re-assigned, ownership is automatically released by MAKASS. If the assign fails, the status of the drive remains un-owned unless the assigning application makes an new explicit ownership request.

24. SETERR - turns on an off display of error messages when there is a net error. Sets variable ERRFLG. Default setting is 0. ,

    ON ENRTY: CL has value to load ERRFLG with.
    0 - error messages off
    ~0 - error messages on

    ON EXIT: ERRFLG is set.

25. GETERR - get the cumulative error count. This baby is incremented whenever there is some kind of net error. Variable is called ERRNUM.

    ON ENTRY: nothing
    ON EXIT: AX contains number of errors.

26. SETTRY - sets variable TRYFLG, which says whether to automatically retry when there is a net error. Default setting is 0FFh.

    ON ENTRY: CL has value to set
    0 - don't retry, ~0 - do retry.

    ON EXIT: TRYFLG is set.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

27. WHO - gets the who info from the master.

ON ENTRY: DX:CX is address of table to load.
ON EXIT: AL has 0 for success, 1 for failure.
IF success the table is filled with WHO info.

FORMAT of WHO INFORMATION:

1 byte - max number of users on system (32 or 64)
16 byte entires * max users - USER ENTRIES
256 bytes (16 bytes * 16 entries) - SPOOL ENTRIES

FORMAT of User Entry:

1 byte - Activity count.
0FFh - active
0 - dead
other - logging out
8 bytes - User Name.
3 bytes - login time (secs,mins,hours)
3 bytes - last request time (secs,mins,hours)
1 byte - last request command byte.

FORMAT of Spool Entry

1 byte - spool status
0 - starting spool
1 - spooling
2 - ready to print
3 - printing
4 - stop print
5 - waiting
0E5h - aborted or done
1 byte - user number of pute spooling.
1 byte - spool ID (top nibble job #, low nibble block #).
2 bytes - time job started (min,hour).
2 bytes - track # of last sector of job.
1 byte - sector # of last sector of job.
8 bytes - spooler's name.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

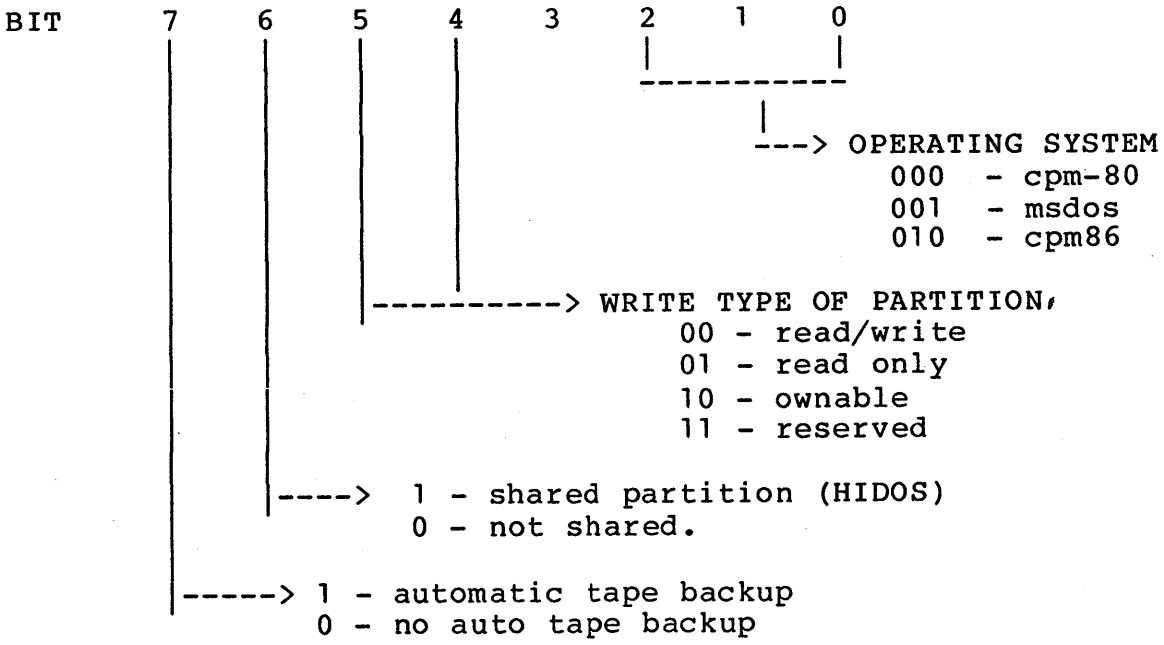
28. DIRNET - gets dirnet info from master (reads alloc table);

ON ENTRY: DX:CX has addr of table to fill out.
ON EXIT: AL has 1 for failure, 0 for success.
On success table is filled with information.

FORMAT of DIRNET INFORMATION: (same as alloc table)

1 byte - size (0 - end of table,1 - 256k, 2 - 512K,
3 - 1MEG, 4 - 2MEG, 5 - 4MEG etc. up to 32MEG)
8 bytes - Partition name.
6 bytes - Partion password. !!!(don't document to public)!!!
1 bytes - Control byte.

FORMAT of CONTROL BYTE -



DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

29. LOCK - used for adding strings to masters lock table.

ON ENTRY: DX:CX has addr of lock string.
ON EXIT: AH and AL tell status of request.

Format of lock string:

1 byte - length of string (max 13)
13 byte - string. String should be blank padded if its less than 13 bytes.

Return status:

reg AL	reg AH	
00	00	- success
01	01h	- denied, owned by someone else
01	81h	- denied, already owned by you
01	83h	- denied, already owned by you, after a retry.
02	02h	- denied, illegal string length
02	82h	- denied, table full
02	03h	- denied, bad transmission (only appears when retry flag is unset).

NOTE: the retry information is given to warn user that the lock may have been successful but we missed the confirmation because of network noise. Then we tried again, but since we have just locked the string it says its locked. It is impossible to tell if it was locked previously, or we successfully locked it this time.

30. UNLOCK - release a locked string in masters lock table.

ON ENTRY: DX:CX has addr of string to release. Format is the same as in LOCK (above).
ON EXIT: AX has status of request, as follows:

reg AL	reg AH	
00	00	- success
01	01h	- denied, locked by someone else
02	02h	- denied, illegal string length
02	82h	- denied, string not found in table
02	83h	- denied, string not found in table after a retry.
02	03h	- denied, bad transmission (only appears when retry flag is unset).

NOTE: Here, as in lock, a garbled response from the master can lead to a retry, and then a spurious deny. This flag warns the user so they can used their best judgement as to how to interpret this.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

31. COERCE - changes NETPAR to 0FFh, causing a flush of disk buffers on the next read or write.

ON ENTRY: nothing
ON EXIT: NETPAR is 0FFh.

32. SETPPA - Sets pointers in BIOS to users poll prime command.

ON ENTRY: DX:CX address of users command block.
If dx = 0 pointers are reset to warm boot values.
ON EXIT: es:bx has address of old command block.

Format of Poll Prime Command Block is:

1 byte - Status byte 00h - station naks poll primes.
01h - station will ack poll primes.
11h - station will ack poll primes and then pass control to user routine addressed in this block. User routine returns in AL this routines response to the poller.
02h (internal) -
2 bytes - reserved
129 bytes - Poll Prime Data
2 bytes - offset of user routine
2 bytes - segment of user routine

33. HDSTAT - gets the hdstat info from the master.

ON ENTRY: DX:CX is address of table to load.
ON EXIT: AL has 0 for success, 1 for failure.
IF success the table is filled with HDSTAT info.

HDSTAT info format:
1 byte - 18h (echoes command for info)
1 byte - ROM version.
1 byte - ROM revision.
1 byte - firmware version number.
1 byte - firmware revision number.
2 bytes - unsed
1 byte - status. 0 - disk present, info OK
~0 - no disk or other error, rest of information invalid.
4 32 byte volume entries.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

Format of Volume Entry -

1 byte	- status.	0FFh - volume present.
		0 - volume not present.
1 byte	- current track for volume.	
1 byte	- number of tracks on volume.	
1 byte	- sectors per track.	
1 byte	- head mask.	
2 bytes	- location of partition offset table.	
2 bytes	- location of bad sector table.	
2 bytes	- location of bad secotr dirty flag.	
1 byte	- error in volume open during firmware init (0 = no error).	
10 bytes	- Volume Label.	
10 bytes	- unused	

34. TIMDAT - gets the time and date info from the master.

ON ENTRY: DX:CX is address of buffer to load.
ON EXIT: AL has 0 for success, 1 for failure.
IF success the buffer is filled with time and date info.

Format of time date info:
62 ticks per sec
One byte each for:
ticks , secs, mins, hours, month, day ,year

35. SETPOL - this module puts bios in automatic poll acking mode. This means that it will answer network polls with an ack and the user should be unaffected, except for time lost in interrupts. SETPOL also enables NMI, in case was disabled by CLRPOL. This routine also polls keyboard to make sure we haven't fallen too far behind while NMI was disabled.

ON ENTRY: nothing
ON EXIT: SIO is reprogrammed, NMI is enabled and we are acking polls.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

36. CLRPOL - disables NMI and waits for a network poll. If no poll comes for 10 miliseconds it calls setpol, which enables NMI and polls keyboard. We start again, looping till approximately 5 second have passed. We then display waiting message and return a failure to user. We will have performed a setpol in this case. If an invalid interrupt is recieved setpol is not called.

ON ENTRY: nothing.
ON EXIT: IF success - carry flag is reset if we have just recieved a poll and not acked.
NMI is disabled.
Automatic acking is disabled.
IF failure - carry flag is set.
no poll was recieved.
AL contains RECSTAT.
NMI may or may not be enabled.
We may or may not be in auto ack mode.

Bit Format of RECSTAT: (7 is high bit)
7 - reset if timeout.
6 - set on CRC error.
5 - set if overrun.
4 - undefined.
3 - set if underrun.
2 - ~~set on error.~~ NO
1 - undefined. error bit on
0 - reset on error.

AL.

37. SNDNET - send data to the master. Can only send after a successful clrpol, so master is waitnig to hear from us. We must do the send imeediately after the clrpol. If we dick around too long the master will timeout waiting for us and move on to the next guy.

ON ENTRY: DX:CX points to send command buffer.
ON EXIT: AX = 0, carry flag reset.
NOTE: this routine assumes the send was successful, as it had no way to tell.

Format of send buffer:
1 byte - network user number.
2 bytes - offset of data buffer to send.
2 bytes - segment of data buffer to send.
2 bytes - length of data buffer to send.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

38. RCVNET - recieve data from the net. This is generally used after a CLRPOL and a SNDNET have made a request from the master.

ON ENTRY: DX:CX points to RECEIVE request buffer.
ON EXIT: IF success - carry flag is reset.
IF failure - carry flag set.
AL has RECSTAT.

For format of RECSTAT see CLRPOL (above).

Format of Receive command buffer:
1 byte - network user number.
2 bytes - offset of data buffer to receive.
2 bytes - segment of data buffer to receive.
2 bytes - length of data buffer to receive.

39. VIDEO (816 ONLY) - this call services all your video output needs. Entries and exits are determined by the function called.

FUNCTIONS are based on value of AH
0 - Set mode - only valid mode B&W 80 x 25
1 - Set cursor type.
2 - Set cursor pos
3 - Read cursor position
4 - Read Light Pen Position (not implemented)
5 - Select Active Page (not implemented)
6 - Scroll active page up
7 - Scroll active page down
8 - Read char/attribute
9 - Write char/attribute
10 - Write char only
11 - Set color palette (not implemented)
12 - write dot (not implemented)
13 - read dot (not implemented)
14 - write char in tty mode
15 - return video status

VIDEO - SET MODE initialzes video controller.
ON ENTRY: AH - 0
ON EXIT: screen is blank. Cursor is a 0,0.

VIDEO - SET CURSOR DISPLAY
Sets cursor display type, top and bottom cell line to display or cursor
On Entry: AH - 1
CH - start line
CL - end line
ON EXIT: nothing

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

VIDEO - SET CURSOR POSITION
On entry: AH - 2
DX - row,col of cursor
ON EXIT: cursor is moved to new position.

VIDEO - GET CURSOR POSITION t cursor postion
On entry: AH - 3
On exit : DX - row and col of cursor.
CX - cursor mode.

VIDEO - SCROLL UP SCREEN
Scrolls the defined window the define number of lines.
On entry: AH - 6
AL - numbers of rows to scroll (0 for blank the area)
CX - row/col of upper left corner
DX - row col of lower right corner
BH - attribute of blanked lines

VIDEO - SCROLL UP DOWN
Scrolls the defined window the define number of lines.
On entry: AH - 7
AL - numbers of rows to scroll (0 for blank the area)
CX - row/col of upper left corner
DX - row col of lower right corner
BH - attribute of blanked lines

VIDEO - READ CHAR and ATTRIBUTE
Reads the character at the current cursor position of screen.
On Entry: AH - 8
On Exit: AL - char
AH - attr

VIDEO - WRITE CHAR and ATTRIBUTE
Writes chars and attributes to screen at current cursor position.
On Entry: AH - 9
CX - count of chars to write
AL - char
BL - attribute
ON EXIT: Cursor position is not changed.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

VIDEO - WRITE CHAR ONLY (no attribute)
Writes chars to screen at current cursor position, not changing the attributes.
On Entry: AH - 10
CX - count of chars to write
AL - char
ON EXIT: Cursor position is not changed.

VIDEO - ,WRITE CHAR IN TTY MODE - outputs a single character with whatever is the current attribute (CURATTR), updating the cursor. It interprets these characters - CR,BS, LF and BELL. IF it gets to end of screen it automatically scrolls. Default screen length is 25 lines, but this can be changed to 24 use SPECIAL CALLS.

On Entry: AH - 11
AL - char to output
ON EXIT: nothing

VIDEO - RETURN VIDEO STATUS
ON ENTRY: AH - 12
On Exit: AH - cols on screen (always 80)
AL - current video mode (always 7)
BH - current active page (always 0)

40. PRINTSCREEN (816 ONLY) - send screen image to printer. Checks flag at 50:0 to see if a screen print is already in progress (1 - in progress). If so it doesn't bother.

ON ENTRY: nothing.
ON EXIT: If success 50:0 hold 0, else 50:0 holds 0FFh.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

41. DMSINFO - gets the DMS INFO table into user buffer.

ON ENTRY: DX:CX is address of buffer to load.
ON EXIT: The buffer is filled with time and date info.
Always succeeds.

FORMAT of DMS INFO table:

1 byte	- product type	
		1 - ZSBC3 machines. (5080,3b,3f,501,3-10X)
		2 - DMS-4
		3 - DMS-1280
		4 - DMS-3C
		5 - DMS-86,5086
		6 - 5016
		7 - 816
		8 - PC card
1 byte	- OS type	11h - cpm-80 (2.2)
		12h - cpm86
		13h - hidos
		21h - mddos (2.1)
4 bytes	- Gutbios Version number	vers,rev,assem,pass
4 bytes	- Prom serial number	
1 byte	- Password mode	0FFh - not auto mode.
		0 thru 9 - auto mode with this PW #.
2 bytes	- Segment address of this table.	
2 bytes	- Offset addres of this table	
1 byte	- Local network user number.	
4 bytes	- Mother bios version number	vers,rev,mod,change

42. WRSCOMMAND - this is used to maek a write status requests from the BIOS and the master.

ON ENTRY: DX:CX points to WRITE STATUS COMMAND BUFFER.
ON EXIT: AL and response status field of COMMAND BUFFER are loaded with network response.

Success	- AL has 04Dh (mesask)
	Command buffer is filled oout
Failure	- other value (currently 04Fh - cmddeny)

NOTE: Success indicates a successfu;l request dialogue, not that the request was granted. You must check Info in the command buffer for that. Failure is generallly due to invalid parmeters in the request (e.g. the partition number doesn't match the partition number in the BIOS part table for the logcial drive number given.)

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

Format of command buffer:

1 byte	- Request type	0 - grant ownership
		1 - release onwership
		2 - force partition to a state
		3 - query currrent status
		4 - release all this users partitions
1 byte	- volume number of partition.	
1 byte	- unit number of partition.	
1 byte	- value.	
1 byte	- physical user	
1 byte	- logical drive (not sent to master)	
1 byte	- response status.	
1 byte	- Unit status.	
8 bytes	- User name.	

WRScommand - entry and exit params for each type of request

GRANT and QUERY request -

ON ENTRY: User must have filled out:
Request type - 0 for grant,3 for query.
Volune number of partition.
Unit number of partition.
Logical drive of partition (A-0,B-1 etc.)

ON EXIT: If request dialogue was successful (response status = 4D):
Unit status has status of drive.
User name has ascii name of drive owner, if there is one.

Unit status values

0FFh	- ownable partiton but not owned.
0FEh	- read/write partition.
0FDh	- read only partition.
0FCh	- hi dos partition. (Can be written to only by HIDOS).
0FBh	- multi. This status is returned on a grant request for a drive you already own it. It may indicate that you own it on another drive.
0FAh	- illegal request.
other	- network user number of current owner o: patition.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

RELEASE request -
ON ENTRY: User must have filled out:
Request type - 1.
Volume number of partition.
Unit number of partition.
Logical drive of partition
(A-0,B-1 etc.)
ON EXIT: If request dialogue was successful
(response status = 4D):
Unit status has status of drive.
User name has ascii name of drive
owner, if there is one.

NOTE: if you attempt to release a partition you don't own returns a failure in response status.

RELEASE ALL request -
ON ENTRY: User must have filled out:
Request type - 4.
ON EXIT: If request dialogue was successful
(response status = 4D):
Write ownership is released an all your
partitions.

FORCE request -
This forces the status of the partition named. You can force the wrtie status on any drive, whether you are assinged to it or not. IF you are not assigned to it set logical drive to 0FFh.

ON ENTRY: User must have filled out:
Request type - 1.
Volume number of partition.
Unit number of partition.
Logical drive of partition
A-0,B-1 etc. if you are
assinged to it.
0FFhif not assigned

ON EXIT: If request dialogue was successful (response
(response status = 4D):
Unit status has status of drive.
User name has ascii name of drive owner, if
there is one.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

43. AUXINIT - FOR 816 ONLY. A service routine for the virtual IBM rombios. It sets parameters for SIO channel A.

ON ENTRY: AL parameter byte.
ON EXIT : paramters are set.

FORMAT of parameter byte:

bit	7	6	5	4	3	2	1	0
	x	x	x	p	p	s	1	1

x bits - don't care (IN IBM used for baud rate which can only be set on 816 with dip switches).
p bits - parity setting
- x0 - no parity
- 01 - odd parity
- 11 - even parity

s bit - stop bits
- 0 - 1 stop bit
- 1 - 2 stop bits

1 bits - character length
- 10 - 7 bits
- 11 - 8 bits

44. PARTADDR - return address of the 86 BIOS Partition Table.

ON ENTRY: DX:CX points to buffer load with addr of PartTab
ON EXIT : Buffer has offset and segment of partitin table loaded into it.

FORMAT: low byte offset, high offset, low segment, high segment

FORMAT of PARTTION TABLE:
- 8 bytes - partition name
- 1 byte - partition size
- 1 byte - unit number
- 1 byte - volume number
- 1 byte - drive type
- 1 byte - write status of drive
- 1 byte - logical drive number (A-0,B-1, etc.)
- 1 byte - media same flag (for MSDOS)

For values of parition size see DIRNET.
For values of drive type see MAKASS.
For values of Write Status see WRScommand.

DESCRIPTION OF GUTBIOS FUNCTIONS
(in numeric order)
(continued)

45. NETINFOREQ - Gets network info into a user buffer.

ON ENTRY: DX:CX points to buffer to load.
ON EXIT : IF failure - AL = 1
IF success - AL = 0
- buffer is laod with net info.

FORMAT of NETINFO:
1 byte - reserved
1 byte - max number of users on system.
126 bytes - unused

46. SETPRESC - This sets the escape codes used for output to a printer attached to console. When outputing to such a device up to a 2 byte escape code is sent before and another after each char is output. This tells the console to pass the char on to the printer. IF a byte in an escape sequence is 0 it marks the end of that sequence.

ON ENTRY: DX:CX points to a buffer with the two sequences in it.
ON EXIT: The sequences are laoded into the BIOS sequence variables.

FORMAT of sequence buffer:
byte one of first sequence
byte two of first sequence
byte one of second sequence
byte two of second sequence

47. SPOOLMESS - this call allows turning on and off console messages from the spool device. Default setting is messages on. Messages should be turned of when you are running a background spooler that you don't want displaying messages in the middle of another application.

ON ENTRY: CL has setting
0 - turn messages off.
~0 - turn message on.
ON EXIT: message display flag (spMessOn) is set.