CONFIDENTIAL - INTERNAL CIRCULATION ONLY

Document PER-2
Title ..... EXDOS - DISKIO Specification
Issue 3
Date ..... 23rd April 1985

CONTENTS
==========

Note: This document describes DISKIO from the point of view of a program which calls it. A separate document (PER-4) contains implementation details to assist in developing the code. Related documents are:

PER-1	EXDOS - System Overview
PER-3	EXDOS - Unit Handler Specification
PER-4	DISKIO and UNITH Implementation Notes
PER-5	EXDOS - System Specification
PER-16	IS-DOS - System Specification

1. INTRODUCTION

DISKIO is the lowest level disk interface code, and as such it communicates directly with the disk controller. The functions it provides are very similar to those provided by the disk code in the IBM ROM (see listing in the IBM technical manual).

It provides reading, writing and verifying of multiple consecutive sectors on a single track, formatting a single track according to a specified track image, reading a complete track image in, reading a sector header and determining the type of a drive. No retries are done on any failed operation, that is the responsibility of the calling program.

The built in unit handler (UNITH) which is described separately (see document PER-3) uses DISKIO for all its accessing of the disk controller. An entry point to DISKIO is provided for the user, so he can access the disks directly, for example the FORMAT program will use the "format track" function of DISKIO. It should never be necessary for any code other than DISKIO to access the disk controller hardware directly.

DISKIO can handle any type of single or double sided soft sectored, single density (FM) and double density (MFM) disks with a data rate of 125K and 250k bits/sec (normally 300rpm) and which use a format compatible with the 1772 disk controller chip. The normal formats which EXDOS uses are single or double sided, double-density, 8 or 9 sectors per track, 512 byte sectors, 40 or 80 tracks per disk. The disk controller and DISKIO are compatible with 3", 3.5" and 5.25" disk drives, with or without ready signals and with or without disk change mechanism.

2. COMMANDS and PARAMETERS

DISKIO has a single entry point at offset 000Dh in the EXDOS ROM. This entry point should be called in Z-80 page-3 with the following parameters in the Z-80 registers:

```
IY = RAM base pointer (see below)
IX = Disk transfer address
 A = Command code - 0 = reset disk system
                    1 = read sectors
                    2 = write sectors
                    3 = verify sectors
                    4 = format track
                    5 = read track
                    6 = read address
                    7 = test disk change
```



```
B = Drive select  (1,2,4,8)    (commands 1...7)
  = Drive mask    (0...15)     (command  0)
                b0 - drive 0
                b1 - drive 1
                b2 - drive 2
                b3 - drive 3
C = Head number   (0,1)        (commands 1,2,3,4,5,6)
D = Track number  (0...79)     (commands 1,2,3,4,5,6)
    0FFh => current track
E = Sector number (1...9)      (commands 1,2,3)
H = b0,1 -  Step rate          (commands 1,2,3,4,5,6)
              0=>6ms   1=>12ms
              2=>20ms  3=>30ms
    b2   -  Precompensation    (commands 2,4)
              1 to enable
    b3   -  Double track flag  (commands 1,2,3,4,5,6)
              1 to enable
    b4   -  0
    b5   -  Recording mode     (commands 1,2,3,4,5,6)
              0=>MFM  1=>FM
    b6,7 -  0
L = Sector count  (1...9)      (commands 1,2,3)
```

The following parameters are returned:

```
Z-flag = Set if no error (A=0)
       = Clear if any error occurred
    A = Error code
          b0 - Not ready          (commands 1,2,3,4,5,6)
          b1 - Verify error       (command  3)
          b2 - Lost data          (commands 1,2,3,4,5,6)
          b3 - CRC error          (commands 1,2,3,6)
          b4 - Record not found   (commands 1,2,3)
          b5 - Disk changed       (commands 1...7)
          b6 - Write protect      (commands 2,4)
          b7 - Unknown command    (commands 8...255)
    B = Disk change support flag  (command 7)
    L = Number of sectors
          successfully transferred (commands 1,2,3)
```

Register AF will be corrupted since it is used to return error flags. All other registers (BC,DE,HL,IX,IY,AF',BC', DE',HL') will be preserved, except where they are used to return results. DISKIO does not check its parameters since this allows the most versatility in using it with non-standard disk formats. However the drive mask (B), head number (C) and step rate (H) will be masked to suitable values before being used, as other values will be meaningless to the hardware of the disk controller.

3. RAM BASE POINTER

DISKIO uses several bytes of variables in the system segment (segment 0FFh) in Z-80 page-2 for storing drive status and so on. These variables are accessed relative to IY and so a pointer to this variable area must be passed to DISKIO in IY, and the system segment must be paged in. When accessed by UNITH, this pointer will already be available in IY since UNITH also uses it.

When the user wants to call DISKIO directly, he must get the correct value for this pointer into IY, and page the system segment into page-2. The value for IY can be obtained from EXDOS by making a scan system extensions call with the command string "EXDOS IY" (see the EXDOS specification - PER-5).

4. DISK TRANSFER ADDRESS

The disk transfer address is required for all data reading, writing or verifying commands. IX must contain the address of the first byte of a contiguous area of memory large enough for the amount of data requested. This memory area must all be available in Z-80 memory space when DISKIO is called since DISKIO does no memory paging at all. Since page-3 must contain the EXDOS ROM, page-2 must contain the system segment and page-0 must contain the page-zero-segment, general purpose disk transfers (such as those done by UNITH) are effectively limited to only accessing page-1, and not crossing segment boundaries. However if a user program is making direct DISKIO calls, it is quite possible to also read or write from page-0, and in this case transfers can cross the page-0/page-1 boundary if desired.

5. RESET DISK SYSTEM COMMAND

This command gives an interrupt to the disk controller chip to reset it if it is busy, and sets the current track number for any drives specified in the drive mask to 0FFh. This ensures that when this drive is next accessed, a seek operation from track zero will be performed to re-locate the track. Normally this command will be issued when a persistent error has been received from a read or write operation, before doing another retry (see ERRORS and RETRY STRATEGY) below. Note that this command never returns an error.

6. READ, WRITE and VERIFY SECTOR COMMANDS

These three commands all take identical parameters (apart from the command number) and return identical results. The only difference is in the direction of the transfer. Verify compares the data at the transfer address with the data on disk and returns a "verify error" as soon as any difference is found.

DISKIO first selects the specified drive, head and recording mode (FM or MFM). Having done this it then checks to see if the disk has changed, if the drive supports the "disk changed" signal (see section 9). If the disk has changed then a "disk changed" error is returned.

It then seeks to the specified track if it is not there already and then looks for the specified sector number. The sector number specifies the first sector from this track which is to be transferred, and the sector count specifies the number of consecutive sectors to be transferred. If the number of sectors requested goes beyond the end of the track then only the number of sectors on the track will be transferred, and a "record not found" error will be returned, there is no automatic stepping.

Normally all sectors are 512 bytes and so this much space must be available at the disk transfer address for each sector requested. However if reading non-standard disk formats the sector size could be 128, 256, 512 or 1024 bytes, or even a mixture of these sizes. It is up to the calling program to ensure that the disk transfer address is large enough, DISKIO does no checking of the size of sectors.

If the "double track" flag is set then the number of stepping pulses issued to get to a given track will be doubled. This enables 40 track disks to be read on 80 track drives.

If the "precompensation" flag is set on a write command then the precompensation facility of the WD1772 chip will be enabled (see data sheet). Normally this should be enabled on the inner tracks and disabled on the outer tracks.

7. FORMAT and READ TRACK COMMANDS

When DISKIO recieves a format track command, it selects the specified drive, head and recording mode, checks for a changed disk, seeks to the specified track if it is not there already and then writes the track image from the user's buffer to the disk. The read track command is similar except that the entire track image is read from the disk to the user's buffer. The nominal number of bytes per track is 6250, however to allow for disk speed variation the disk transfer address should be larger than this, preferably 6600 bytes.

Note that the data read or written with these commands includes the index mark, sector headers, gaps, address marks and CRC's as well as the data bytes from the sectors. In particular when writing a track it is essential that the correct bytes are included to control the formatting of sectors etc. Details of this can be found in the data sheet for the 1772 disk controller chip.

8. READ ADDRESS COMMAND

This command will normally be called after an error has occurred to assist the calling program (particularly UNITH) in determining the error. If the track specified is 0FFh then it reads the next sector header from whichever track the head happens to be on. Otherwise it does a normal seek operation before reading the sector header. Six bytes are read into the disk transfer address according to the following table:

IX+0	Track number (0...255)	
IX+1	Side number (0 or 1)	
IX+2	Sector number (0...255)	
IX+3	Sector length (0...3)	0 => 128 bytes 1 => 256 bytes 2 => 512 bytes 3 => 1024 bytes
IX+4,5	CRC bytes	

9. TEST DISK CHANGE COMMAND

The test disk change command is just given a drive mask which must select exactly one drive. It returns the same bit set if this drive supports the "disk changed" signal and the whole byte zero if not. If the drive does support the disk changed signal then this command may also return a "disk changed" error if the signal was actually set, and it will be reset.

Note that an implicit TEST DISK CHANGE command is done whenever any other read, write or verify command is given to DISKIO, so these commands can all return a "disk changed" error. In these cases also the disk changed signal is reset when it has been read so the next command will not return the error again.

10. ERRORS and SUGGESTED RETRY STRATEGY

When DISKIO gets an error from the disk controller hardware, it is returned to the caller in register A. In general when an error is returned it is recomended that the calling program should retry the operation several times before giving up and reporting the error to the user, although the details vary a bit between errors. The user should always be given the option of retrying the operation since it may be something which he can remedy (by putting a disk in the drive for example).

If a NOT READY or WRITE PROTECT error occurs, the operation should be retried once and if this fails then it should be reported to the user. DISKIO carries out the necessary waiting for motor start up etc. and so retrying these errors more times will not help without user intervention.

A LOST DATA error should not occur since it implies that the internal DISKIO code was not fast enough. However if it does occur it should be treated like a CRC error.

CRC or RECORD NOT FOUND are the most likely errors and normally indicate a soft data error which can be recovered by retrying. If one of these errors occurs it should first be retried immediately, if this fails it should be retried two or three more times, with a "reset disk system" call between each retry to reset this disk drive and so force it to re-seek the track. If this still fails then the error should be reported to the user.

With a RECORD NOT FOUND error, it may be worth doing a READ ADDRESS command to see where the head is on the disk. If the track number is wrong, or if the READ ADDRESS command fails then it could be that the "double track" flag was set wrongly.

A verify error can only occur from a "verify sectors" command and is detected by the DISKIO software rather than by the disk controller hardware. If it occurs then it should be retried as with CRC errors, but it may be desirable to warn the user about the fault even if the retry succeeds since it could mean that the disk is suspect and could fail soon.

++++++++++ END OF DOCUMENT ++++++++++

