BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
Study Material

As soon as power is switched on, CPU starts executing the assembly code located at F000:FFF0. This area contains the BIOS, Basic Input/Output System. This code will perform Power On Self Test (POST) where it checks for installed devices like video card, hard drives, etc and checks if they all work. Finally it transfers control to the operating system. After the BIOS have successfully completed the POST it calls interrupt 19h. INT 19h attempts to read in the Boot Sector of the 1st floppy disk. If it fails, it does the same thing on the 1st hard drive. The sequence of drives to be checked while booting can be set in the BIOS features. If the check fails it returns an error message. It checks the 16-bit word at absolute address 07DFEh for AA55h. This is the boot signature and BIOS uses this to ensure that the sector contains a valid boot sector. When the BIOS find the valid boot sector, it reads that sector (512 bytes) off of the disk into memory at 0000:7C00h and interrupt 19h jumps there to start executing the code. Thus boot sector code gets control and DL will be loaded with

00h if the boot sector was loaded from drive A,
80h if the boot sector was loaded from drive C

This way, the boot sector can determine which drive it was booted from.

Logical sector 0 known as the boot sector contains all the critical information regarding the disk medium's characteristics. The table in the next page gives the details about the boot sector. The first byte in this sector should be a jump instruction. As shown in the table, the first byte should be either a intrasegment jump JMP (opcode 0E9H) followed by a 16 bit immediate value (displacement) or a short jump JMP (opcode 0EBH) followed by an 8-bit immediate value (displacement) and then by an no-operation NOP (opcode 90h). The absence of jump instruction indicates that the disk has not been formatted or was not been formatted for the use with MS-DOS. The jump instruction itself does not ensure that the disk has been formatted.

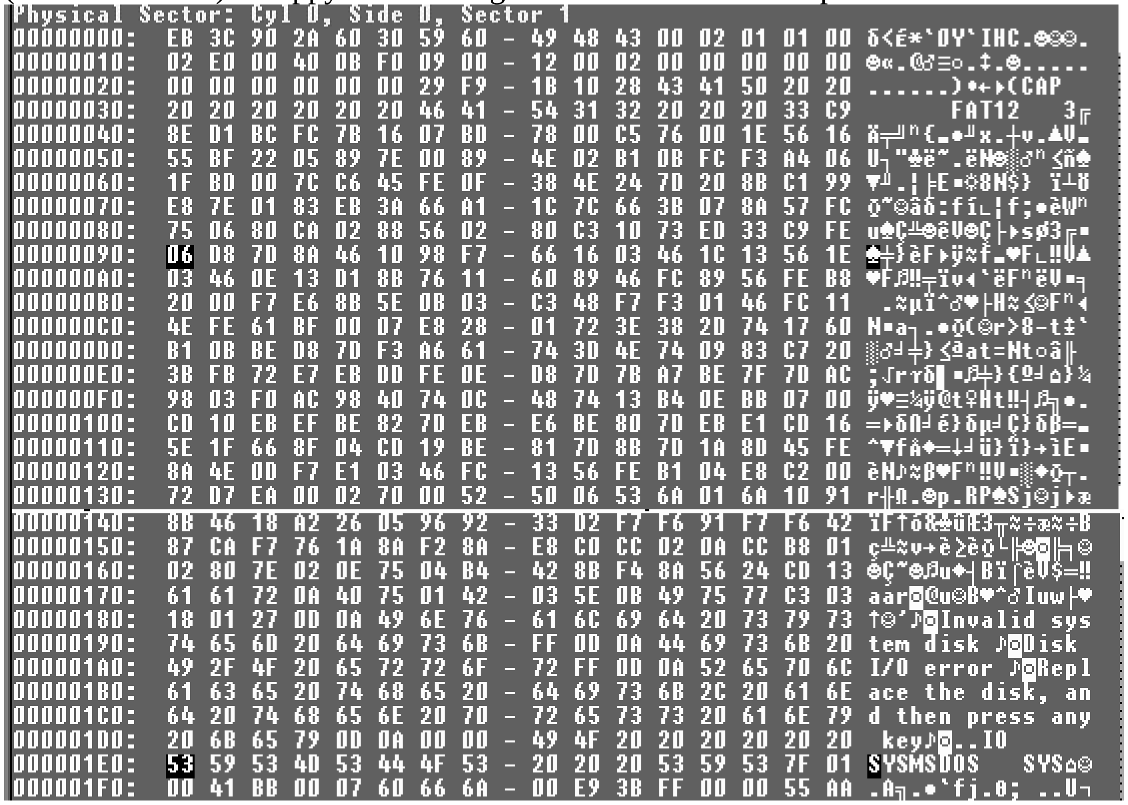
After the initial JMP instruction there is an 8-byte field reserved by Microsoft for OEM identification. This field contains the disk-formatting program, with the name of the manufacturer and the manufacturer's internal MS-DOS version number. This is different for each brand of computer, disk controller, and medium.

The third and most important part of boot sector is bios parameter block (BPB). It contains the information useful for calculating the addresses for various sectors in the logical volume. BPB also includes the information about FAT and root directories. The final component of the boot sector is bootstrap.

00 - 02H (3 bytes)	E9 XX XX or EB XX 90
03 - 0AH (8 bytes)	Disk formatting program
0B - 0CH (2 bytes)	Number of bytes per sector
0DH (1 byte)	Number of sectors per Cluster
0E - 0FH (2 bytes)	Reserved sectors (including the boot sector itself)
10H (1 byte)	Number of File Allocation Tables (FAT)
11 - 12H (2 bytes)	Number of root directory entries
13 - 14H (2 bytes)	Total sectors on disk
15H (1 byte)	Media descriptor byte
16 - 17H (2 bytes)	Number of sectors per FAT
18 - 19H (2 bytes)	Sectors per track
1A - 1BH (2 bytes)	Number of heads
1C - 1FH (4 bytes)	Number of hidden sectors
20 - 23H (4 bytes)	Total sectors in logical volume (volume size > 32 MB)
24H (1 byte)	Physical drive number
25H (1 byte)	Reserved
26H (1 byte)	Extended boot signature record (29H)
27 - 2AH (4 bytes)	Volume serial number
2B - 35H (11 bytes)	Volume label
36 - 3DH (8 bytes)	Reserved
3EH-	Bootstrap

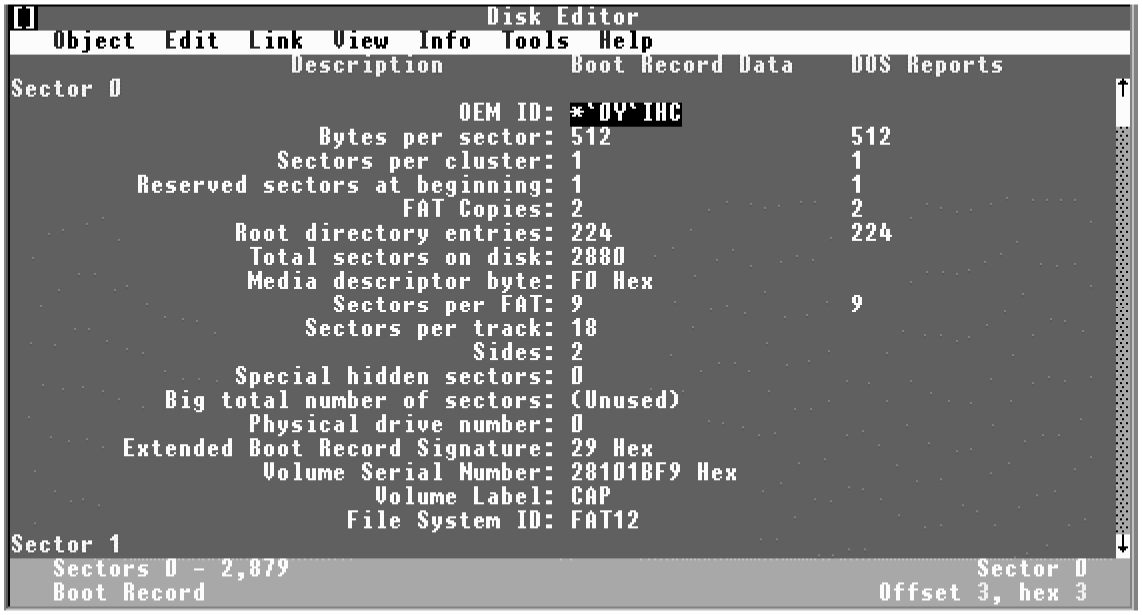
Table: BOOT SECTOR

As an example we present the boot sector of a formatted floppy disk. First we show the hex code pattern followed by the logical pattern. You can also do the same by formatting (if not formatted) a floppy and viewing the sector 0 with the help DISKEDIT.



```
Physical Sector: Cyl 0, Side 0, Sector 1
00000000:  EB 3C 90 2A 60 30 59 60 - 49 48 43 00 02 01 01 00 δ<É*`0Y`IHC.☻☺☺.
00000010:  02 E0 00 40 0B F0 09 00 - 12 00 02 00 00 00 00 00 ☻α.@♂≡○.‡.☻.....
00000020:  00 00 00 00 00 00 29 F9 - 1B 10 28 43 41 50 20 20 ......)·←►(CAP
00000030:  20 20 20 20 20 20 46 41 - 54 31 32 20 20 20 33 C9       FAT12   3╔
00000040:  8E D1 BC FC 7B 16 07 BD - 78 00 C5 76 00 1E 56 16 Ä╤╝ⁿ{▬•╜x.┼v.▲V▬
00000050:  55 BF 22 05 89 7E 00 89 - 4E 02 B1 0B FC F3 A4 06 U┐"♣ë~.ëN☻▒♂ⁿ≤ñ♠
00000060:  1F BD 00 7C C6 45 FE 0F - 38 4E 24 7D 20 8B C1 99 ▼╜.|╞E■☼8N$} ï┴Ö
00000070:  E8 7E 01 83 EB 3A 66 A1 - 1C 7C 66 3B 07 8A 57 FC Φ~☺âδ:fí∟|f;•èWⁿ
00000080:  75 06 80 CA 02 88 56 02 - 80 C3 10 73 ED 33 C9 FE u♠Ç╩☻êV☻Ç├►sφ3╔■
00000090:  06 D8 7D 8A 46 10 98 F7 - 66 16 03 46 1C 13 56 1E ♠╪}èF►ÿ≈f▬♥F∟‼V▲
000000A0:  03 46 0E 13 D1 8B 76 11 - 60 89 46 FC 89 56 FE B8 ♥F♫‼╤ïv◄`ëFⁿëV■╕
000000B0:  20 00 F7 E6 8B 5E 0B 03 - C3 48 F7 F3 01 46 FC 11  .≈µï^♂♥├H≈≤☺Fⁿ◄
000000C0:  4E FE 61 BF 00 07 E8 28 - 01 72 3E 38 2D 74 17 60 N■a┐.•Φ(☺r>8-t↨`
000000D0:  B1 0B BE D8 7D F3 A6 61 - 74 3D 4E 74 09 83 C7 20 ▒♂╛╪}≤ªat=Nt○â╟ 
000000E0:  3B FB 72 E7 EB DD FE 0E - D8 7D 7B A7 BE 7F 7D AC ;√rτδ▌■♫╪}{º╛⌂}¼
000000F0:  98 03 F0 AC 98 40 74 0C - 48 74 13 B4 0E BB 07 00 ÿ♥≡¼ÿ@t♀Ht‼┤♫╗•.
00000100:  CD 10 EB EF BE 82 7D EB - E6 BE 80 7D EB E1 CD 16 ═►δ∩╛é}δµ╛Ç}δß═▬
00000110:  5E 1F 66 8F 04 CD 19 BE - 81 7D 8B 7D 1A 8D 45 FE ^▼fÅ♦═↓╛ü}ï}→ìE■
00000120:  8A 4E 0D F7 E1 03 46 FC - 13 56 FE B1 04 E8 C2 00 èN♪≈ß♥Fⁿ‼V■▒♦Φ┬.
00000130:  72 D7 EA 00 02 70 00 52 - 50 06 53 6A 01 6A 10 91 r╫Ω.☻p.RP♠Sj☺j►æ
00000140:  8B 46 18 A2 26 D5 96 92 - 33 D2 F7 F6 91 F7 F6 42 ïF↑ó&╒ûÆ3╥≈÷æ≈÷B
00000150:  87 CA F7 76 1A 8A F2 8A - E8 C0 CC 02 0A CC B8 01 ç╩≈v→è≥èΦ└╠☻◙╠╕☺
00000160:  02 80 7E 02 0E 75 04 B4 - 42 8B F4 8A 56 24 CD 13 ☻Ç~☻♫u♦┤Bï⌠èV$═‼
00000170:  61 61 72 0A 40 75 01 42 - 03 5E 0B 49 75 77 C3 03 aar◙@u☺B♥^♂Iuw├♥
00000180:  18 01 27 0D 0A 49 6E 76 - 61 6C 69 64 20 73 79 73 ↑☺'♪◙Invalid sys
00000190:  74 65 6D 20 64 69 73 6B - FF 0D 0A 44 69 73 6B 20 tem disk ♪◙Disk 
000001A0:  49 2F 4F 20 65 72 72 6F - 72 FF 0D 0A 52 65 70 6C I/O error ♪◙Repl
000001B0:  61 63 65 20 74 68 65 20 - 64 69 73 6B 2C 20 61 6E ace the disk, an
000001C0:  64 20 74 68 65 6E 20 70 - 72 65 73 73 20 61 6E 79 d then press any
000001D0:  20 6B 65 79 0D 0A 00 00 - 49 4F 20 20 20 20 20 20  key♪◙..IO
000001E0:  53 59 53 4D 53 44 4F 53 - 20 20 20 53 59 53 7F 01 SYSMSDOS   SYS⌂☺
000001F0:  00 41 BB 00 07 60 66 6A - 00 E9 3B FF 00 00 55 AA .A╗.•`fj.Θ; ..U¬
```

fig: hex code of boot sector of a formatted floppy.



```
[]                                  Disk Editor
   Object   Edit   Link   View   Info   Tools   Help
                       Description          Boot Record Data     DOS Reports
Sector 0
                                  OEM ID: *`0Y`IHC
                        Bytes per sector: 512                   512
                     Sectors per cluster: 1                     1
              Reserved sectors at beginning: 1                  1
                              FAT Copies: 2                     2
                 Root directory entries: 224                    224
                  Total sectors on disk: 2880
                  Media descriptor byte: F0 Hex
                        Sectors per FAT: 9                      9
                      Sectors per track: 18
                                  Sides: 2
                 Special hidden sectors: 0
             Big total number of sectors: (Unused)
                  Physical drive number: 0
           Extended Boot Record Signature: 29 Hex
                   Volume Serial Number: 28101BF9 Hex
                           Volume Label: CAP
                         File System ID: FAT12
Sector 1
   Sectors 0 - 2,879                                              Sector 0
   Boot Record                                              Offset 3, hex 3
```

fig: logical interpretation boot sector of a formatted floppy.

ROOT DIRECTORY

Root Directory occupies the area below FAT. The directory contains 32-byte entries that contain information about files, and other directories. The table given below gives the information about the various fields in the 32-byte entries.

00 - 07H (8 bytes)	Filename
08 - 0AH (3 bytes)	File Extension
0BH (1 byte)	File Attribute
0C – 15H (9 bytes)	Reserved
16 - 17H (2 bytes)	Time of creation or Last Updating
18 - 19H (2 bytes)	Date of creation or Last Updating
1A - 1BH (2 bytes)	Starting Cluster Number
1C – 1FH (4 bytes)	Size of the File

Table: 32-byte Root Directory entry

If the first byte of the directory entry is one among the values given below, then they are interpreted as follows

Content	**Interpretation**
E5H	Erased File
00H	A Directory Entry which has never been used or the end of occupied portion of the Directory
05H	First character of the file name is E5H
2EH	Current or Parent directory entry. The presence of 2EH in the next byte represents that the cluster number field contains the cluster number of the parent directory.

The bits in the attribute byte are interpreted as follows

Bit	**Interpretation**
0	Read only file
1	Hidden file
2	System file
3	Volume label
4	Directory
5	Archive bit
6	Reserved
7	Reserved

The time field can be interpreted as follows

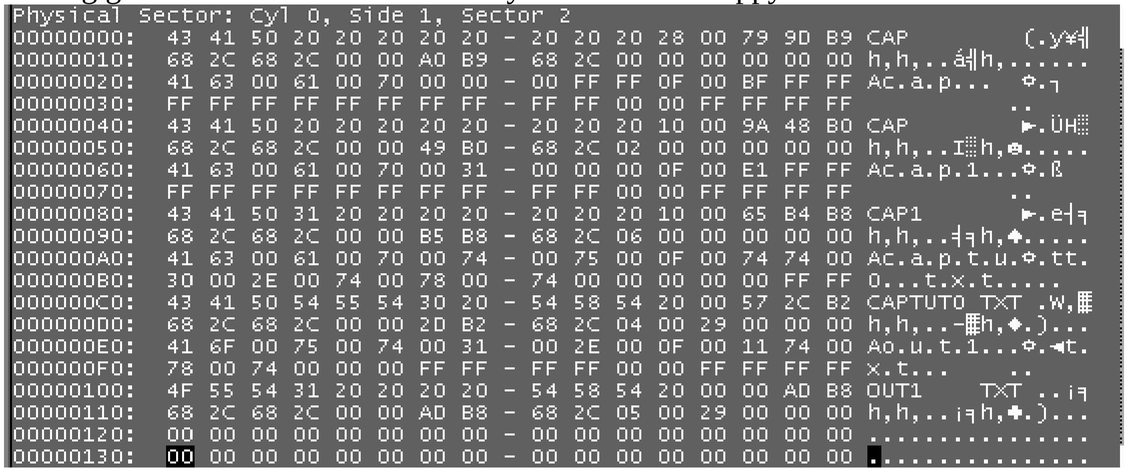
Bits	**Contents**
0-4	Binary Numbers from 0-29 corresponding to 0-58 seconds
5-10	Binary Numbers from 0-59 corresponding to 0-59 minutes
11-15	Binary Numbers from 0-23 corresponding to 0-23 hours

The date field can be interpreted as follows

Bits	**Contents**
0-4	Day(1-31)
5-8	Month(1-12)
9-15	Year(Base year 1980)

The size of the file field is interpreted as 4-byte integer with the lower two bytes of the number stored first.

The size and position of the root directory are fixed and are decided by the formatting program.
Now we take an example show how the root directory entries are and how they are interpreted.
The fig given below is the root directory taken from a floppy.



```
Physical Sector: Cyl 0, Side 1, Sector 2
00000000:  43 41 50 20 20 20 20 20 - 20 20 20 28 00 79 9D B9 CAP        (.y¥╣
00000010:  68 2C 68 2C 00 00 A0 B9 - 68 2C 00 00 00 00 00 00 h,h,..á╣h,......
00000020:  41 63 00 61 00 70 00 00 - 00 FF FF 0F 00 BF FF FF Ac.a.p...  ☼.┐
00000030:  FF FF FF FF FF FF FF FF - FF FF 00 00 FF FF FF FF           ..
00000040:  43 41 50 20 20 20 20 20 - 20 20 20 10 00 9A 48 B0 CAP        ►.ÜH░
00000050:  68 2C 68 2C 00 00 49 B0 - 68 2C 02 00 00 00 00 00 h,h,..I░h,☻.....
00000060:  41 63 00 61 00 70 00 31 - 00 00 00 0F 00 E1 FF FF Ac.a.p.1...☼.ß
00000070:  FF FF FF FF FF FF FF FF - FF FF 00 00 FF FF FF FF           ..
00000080:  43 41 50 31 20 20 20 20 - 20 20 20 10 00 65 B4 B8 CAP1       ►.e┤╕
00000090:  68 2C 68 2C 00 00 B5 B8 - 68 2C 06 00 00 00 00 00 h,h,..╡╕h,♠.....
000000A0:  41 63 00 61 00 70 00 74 - 00 75 00 0F 00 74 74 00 Ac.a.p.t.u.☼.tt.
000000B0:  30 00 2E 00 74 00 78 00 - 74 00 00 00 00 00 FF FF 0...t.x.t.....
000000C0:  43 41 50 54 55 54 30 20 - 54 58 54 20 00 57 2C B2 CAPTUT0 TXT .W,▓
000000D0:  68 2C 68 2C 00 00 2D B2 - 68 2C 04 00 29 00 00 00 h,h,..-▓h,♦.)...
000000E0:  41 6F 00 75 00 74 00 31 - 00 2E 00 0F 00 11 74 00 Ao.u.t.1...☼.◄t.
000000F0:  78 00 74 00 00 00 FF FF - FF FF 00 00 FF FF FF FF x.t...     ..
00000100:  4F 55 54 31 20 20 20 20 - 54 58 54 20 00 00 AD B8 OUT1    TXT ..¡╕
00000110:  68 2C 68 2C 00 00 AD B8 - 68 2C 05 00 29 00 00 00 h,h,..¡╕h,♣.)...
00000120:  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 ................
00000130:  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 ................
```

Now we decode the entry from 0000 to 001F H.
The first byte does not contain any special information.
The first field of the root directory (Filename) is from 00 to 07H .In this case it is CAP.
There is no extension for the file (08 to 0AH).
The attribute contains 28H which means the bits 3and 5 are set.
The time field has A0H in first byte and B9H in second byte.
Note: The word has to be interpreted as B9A0H because the lower byte is stored first.
Binary representation of B9A0H is 1 0 1 1 1 0 0 1 1 0 1 0 0 0 0 0

Bits 0-4 has 0 which implies zero seconds (remember to multiply the binary number with two).

Bits 5-10 have 13 which imply 13 minutes.
Bits 11-15 have 23 which imply 23 hours.

The date field has the word 2C68H.
Binary representation of 2C68H is 0 0 1 0 1 1 0 0 0 1 1 0 1 0 0 0

Bits 0-4 have 8 which implies the day is 8th.
Bits 5-8 have 3 which implies the month march.
Bits 9-15 have 22 which implies 22nd year from 1980 i.e. 2002.

Hence the file was created or last updated at 23hr 13min 0sec on March 8, 2002

Next is the starting cluster number of the file which in this case is 0 (bytes 1A-1BH).
The file size for this file is 0.
Here you should know how the four byte integer is stored. Suppose we have 2A 3B 4C 5D in the file size field then the actual hex number is 5D4C3B2AH.

After the root directory the area is The Files Area.

In case of 1.44 MB floppy in logical volume is divided as follows.

Sector 0	Boot Sector
Sectors 1 to 9	I st FAT area
Sectors 10 to 18	2 nd FAT area
Sectors 19 to 32	Root Directory Area
Sectors 33 to 2879	The Files Area

Logical partition of a MS-DOS disk
The first part of the disk is Boot Sector as we discussed earlier. Following that is File Allocation Table. Then there may be copies of FAT. This followed by the Root Directory and Files Area. The logical partition is given below.

Boot sector
FAT #1
Additional copies of FAT
Root directory
Files area

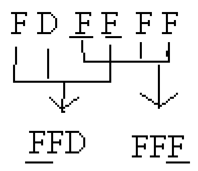
File Allocation Table (FAT)

File Allocation Table contains the information of the group of sectors that are assigned to the file, when it is created. Generally group of sectors is power of 2. These groups of sectors are known as allocation units or clusters. Number of sectors per allocation unit or cluster can be found out from the Bios Parameter Block (BPB) present in the boot sector at an offset of 0Dh.
The fields in the FAT give the cluster number that is allocated to the file. These fields can be 12 bits or 16 bits depending on the size of the medium (12 bits if the disk contains fewer than 4087 clusters). The first two entries in the FAT are always reserved. Hence the assignable cluster always starts from the 2nd cluster.

12 bit content	16 bit content	Interpretation
000H	0000H	Cluster is available
FF0 – FF6 H	FFF0 – FFF6 H	Cluster is reserved
FF7H	FFF7H	Bad Cluster
FF8-FFF H	FFF8-FFFF H	Last Cluster in the File
Any other value	Any other value	Next Cluster in the File

The root directory entry of a file contains the starting cluster number of that file, which is the entry point into the FAT. From then, each FAT field contains the pointer to the next cluster of the file, till we face the last cluster (FF8-FFF h). Usually an identical copy of the FAT is maintained in the floppy. Whenever the files are updated, all the FAT copies will be changed accordingly. The additional FAT copies can be used when the first FAT copy becomes unreadable.

Interpreting the File Allocation Table
Here we take an example of 32-byte FAT entry and decode it to show how a file will be allocated clusters in the memory. In order to represent a cluster we need at least 12 bits. But all the data is byte organized in the memory. So consecutive 3 bytes can be interpreted as two cluster numbers in the following way. Remember that the most significant byte of a number is stored after the lower order byte.



The 32-byte FAT entry can be decoded as follows

```
FD FF FF   FF 4F 00   FF 8F 00   07 F0 FF   09 A0 00   0B E0 00   FF FF FF   0F 00 01   11 20 01   13 F0 FF
FFD  FFF   FFF  004   FFF  008   007  FFF   009   00A  00B  00E   FFF   FFF  00F  010   011  012   013  FFF
```

This can be interpreted as

Starting cluster number from the root directory entry

Cluster number	Next cluster number
0	FFD
1	FFF
2	FFF
3	004
4	FFF
5	008
6	007
7	FFF
8	009
9	00A
0A	00B
0B	00E
0C	FFF
0D	FFF
0E	00F
0F	010
10	011
11	012
12	013
13	FFF

Starting from cluster number 5 the next cluster number allocated to file is 008. After going to cluster number 8 we find that the next cluster number that is allocated to the file is 009 as represented in the table. Further, we see that the cluster number 9 field has 00A as the next cluster number. In this way, we proceed till we face the end of the file represented by FFF in next cluster number field (last cluster in file).

****