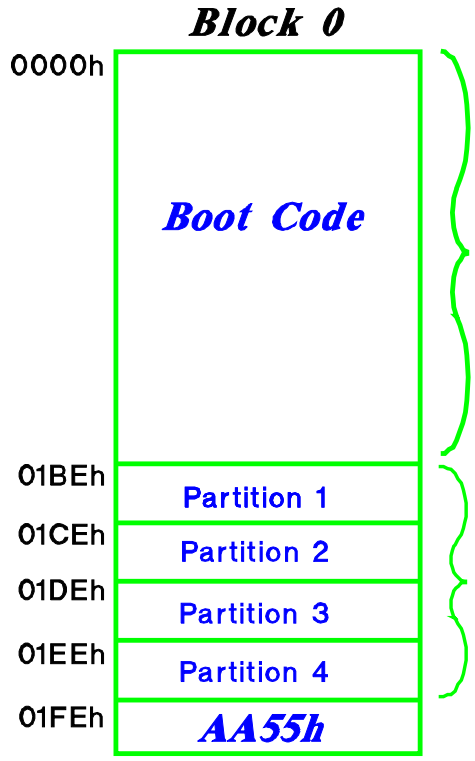
Master Boot Record

Block zero on the disk



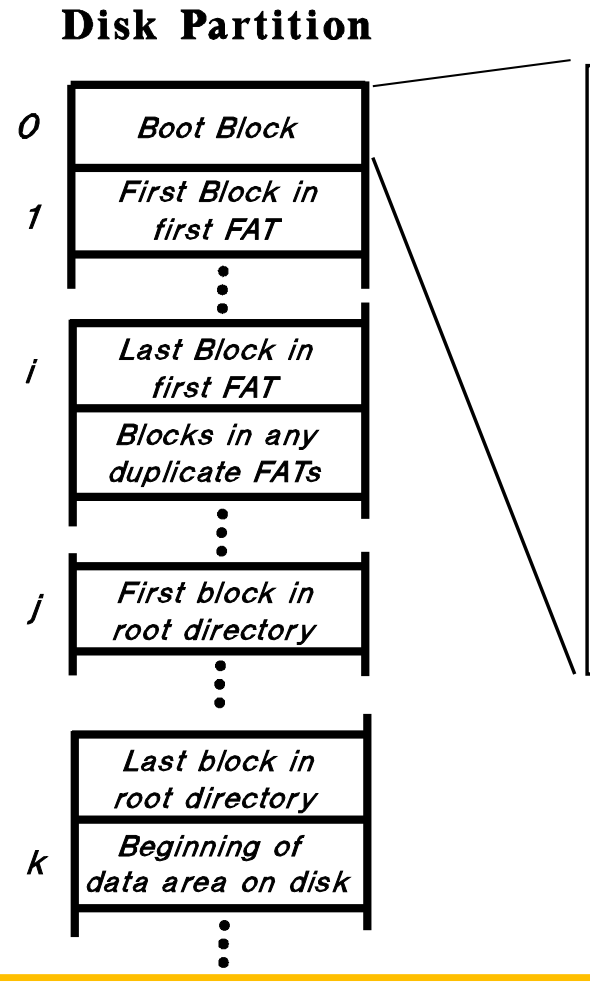
0. BIOS transfers to location 0000h
1. Locate "active" partition
2. Load 1st sector of partition into memory
3. Transfer execution to that code

Partition Table

Signature

DISK0010

DOS Partition

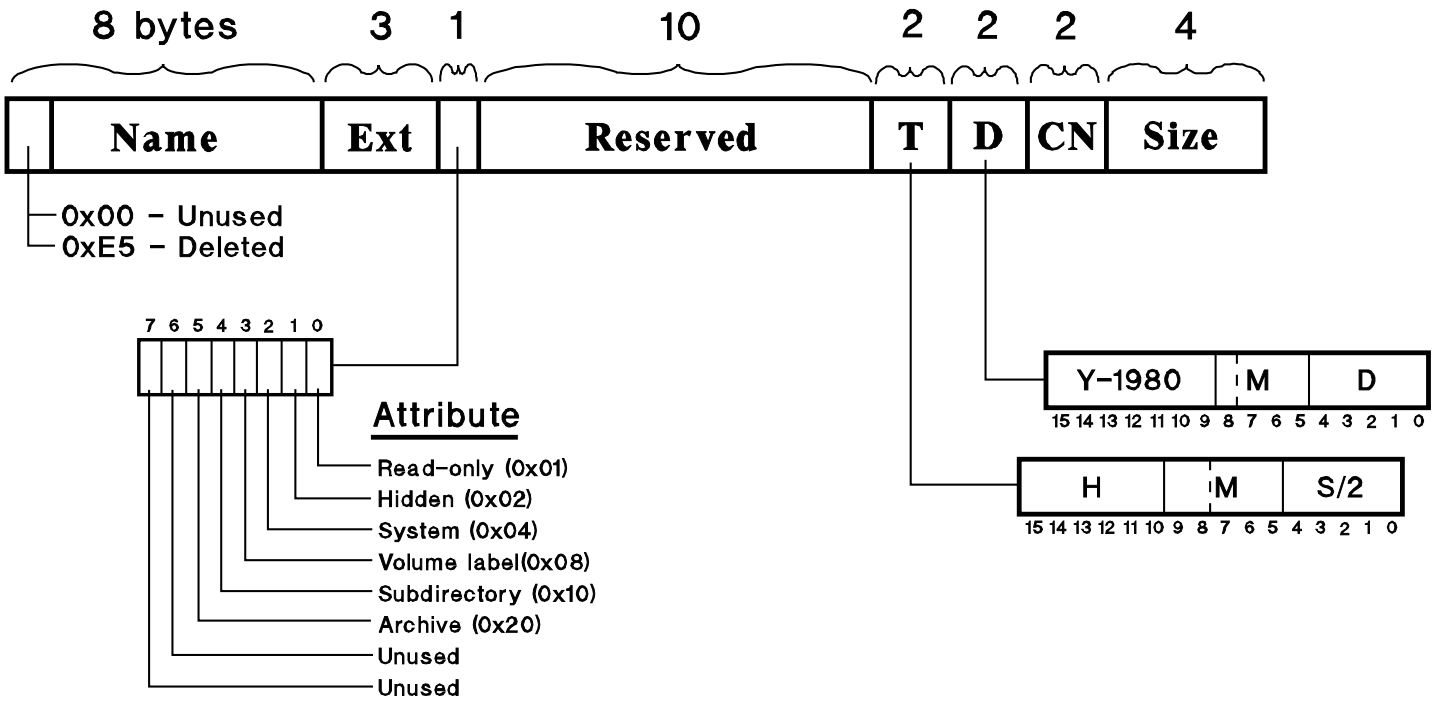


Boot Block Contents

0x00	0x02	<jump to bootstrap>
0x03	0x0a	Computer mfg name
0x0b	0x0c	Bytes per block (bpb)
0x0d		Blocks/cluster (bpc)
0x0e	0x0f	Reserved blocks (for boot record) (rb)
0x10	0x10	# of FATs (nFAT)
0x11	0x12	# of root directory entries (nerd)
0x13	0x14	# of logical blocks
0x15	0x15	Medium Descriptor
0x16	0x17	# of blocks/FAT (bpf)
0x18	0x19	Sectors/track
0x1a	0x1b	# of heads (surfaces)
0x1c	0x1d	# of hidden blocks (hb)
0x1e	....	Bootstrap program

DISK0020

DOS Directory Entry



8 entries per block

DISK0030

FAT Values

0x000 – Unused block

0xFF0 – 0xFF6 – Reserved

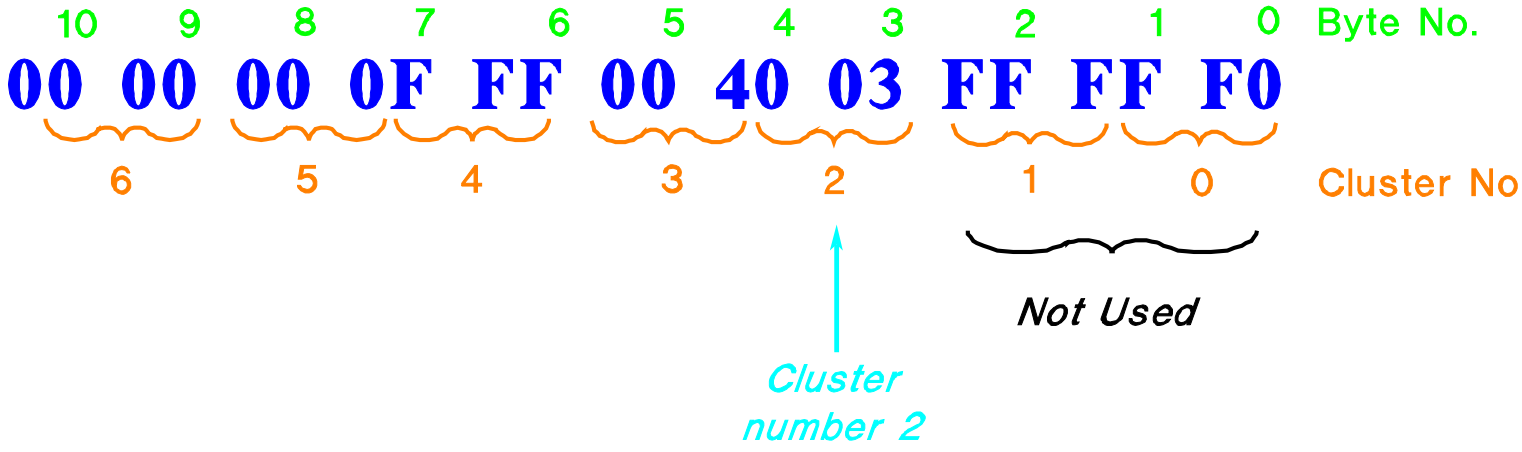
0xFF7 – Bad Cluster

0xFF8 – 0xFFF – Last cluster in file

Anything else – next cluster in file

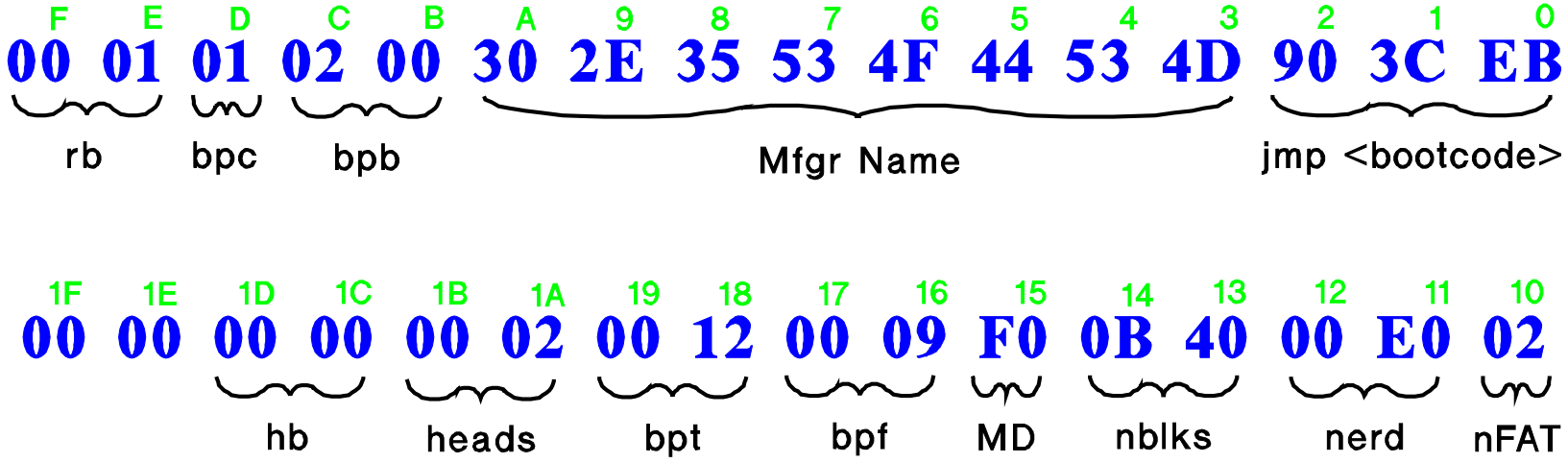
12 Bit FAT Example

The following is the FAT table for a 1200 byte file, 1 block/cluster, 512 bytes/block, starting at cluster number 2. DOS values are stored "little endian."



DISK0050

Boot Sector Example



DISK0060

Some Equations

Location of first FAT

FATstrt = hb + rb

Location of i-th FAT

FATstrt[i] = hb + rb + i*bpf

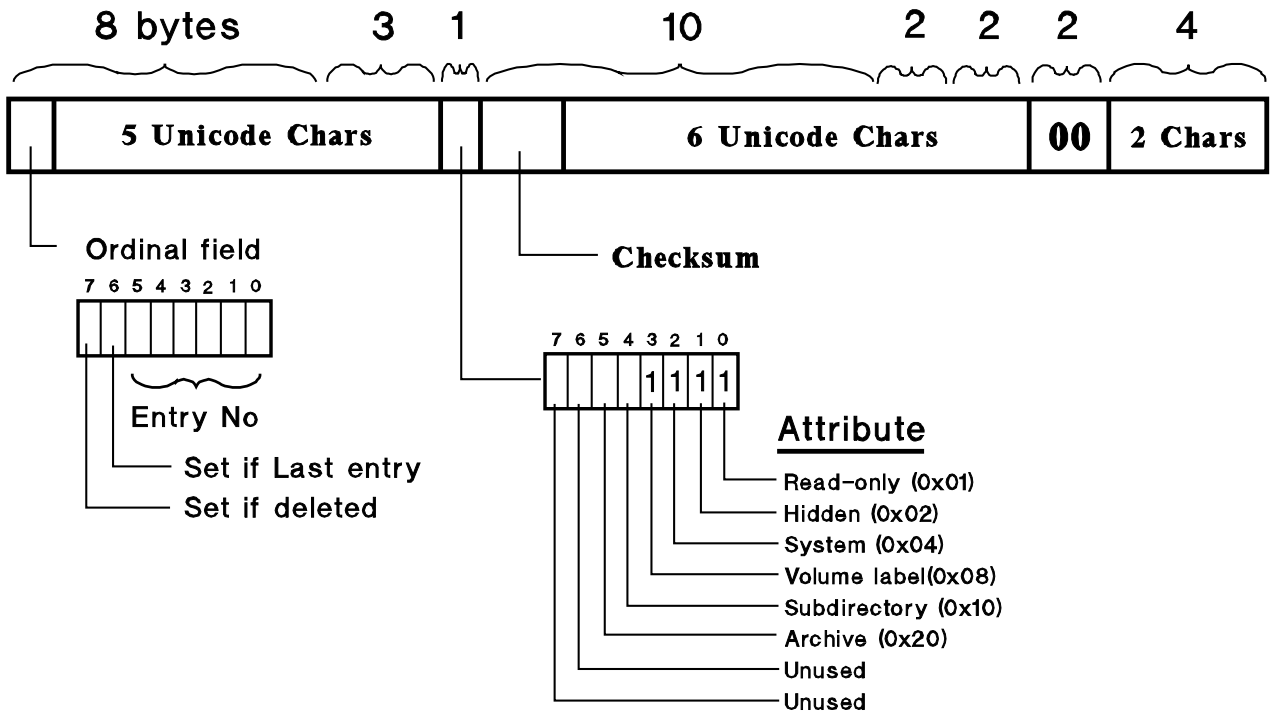
Number of blocks in root directory

nRoot = (nerd*32 + bpb - 1) / bpb

Logical block number of first block in a cluster

lbn = hb + rb + nFAT*bfp + nRoot + (cln -2)*bpc

VFAT Long File Names



DISK0080

Example Long File Name