

```
1 CS5600, Handout Week10.b
2
3 // Code snippets borrowed from WeensyOS boot loader
4 // They illustrate how kernel "talks" to a disk through programmed I/O:
5 // the boot loader reads in the kernel from the disk.
6 //
7 // See the functions boot_waitdisk() and boot_readsect(). Compare to
8 // Figures 36.5 and 36.6 in 0STEP.
9
10
11 // WeensyOS boot loader loads the kernel at address 0x40000 from
12 // the first IDE hard disk.
13 //
14 // A BOOT LOADER is a tiny program that loads an operating system into
15 // memory. It has to be tiny because it can contain no more than 510 bytes
16 // of instructions: it is stored in the disk's first 512-byte sector.
17
18 #define SECTORSIZE      512
19
20 // boot_readsect(dst, src_sect)
21 //   Read disk sector number `src_sect` into address `dst`.
22 static void boot_readsect(uintptr_t dst, uint32_t src_sect) {
23     // programmed I/O for "read sector"
24     boot_waitdisk();
25     outb(0x1F2, 1);           // send `count = 1` as an ATA argument
26     outb(0x1F3, src_sect);    // send `src_sect`, the sector number
27     outb(0x1F4, src_sect >> 8);
28     outb(0x1F5, src_sect >> 16);
29     outb(0x1F6, (src_sect >> 24) | 0xE0);
30     outb(0x1F7, 0x20);       // send the command: 0x20 = read sectors
31
32     // then move the data into memory
33     boot_waitdisk();
34     insl(0x1F0, (void*) dst, SECTORSIZE/4); // read 128 words from the disk
35 }
36
37 // boot_waitdisk
38 //   Wait for the disk to be ready.
39 static void boot_waitdisk(void) {
40     // Wait until the ATA status register says ready (0x40 is on)
41     // & not busy (0x80 is off)
42     while ((inb(0x1F7) & 0xC0) != 0x40) {
43         /* do nothing */
44     }
45 }
46
47 }
```

Hard disk drive organization

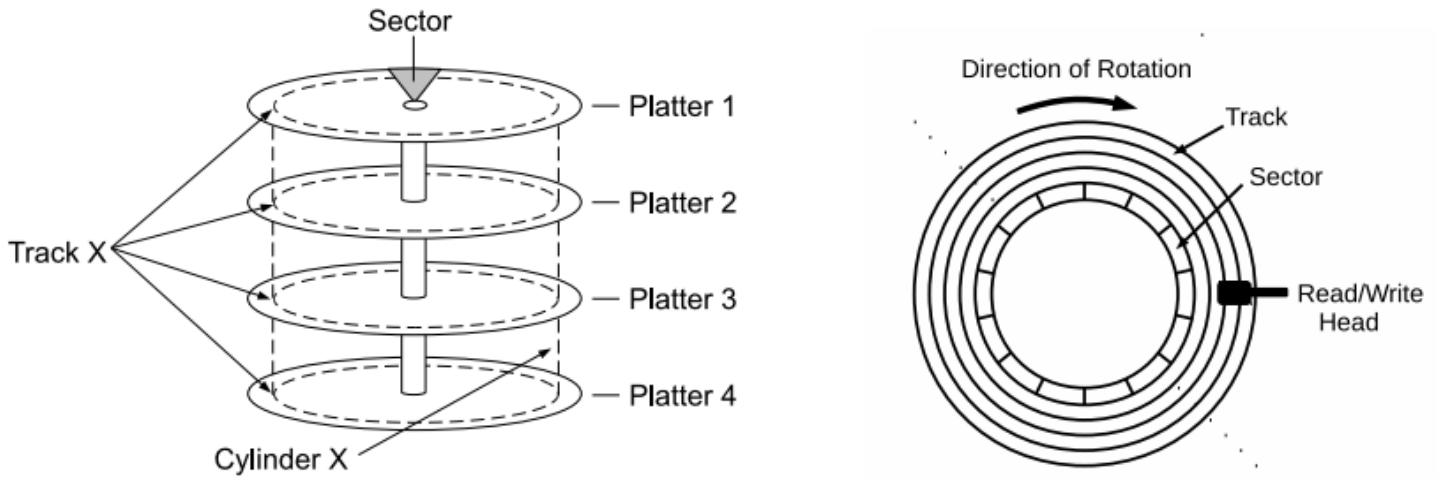


Fig 5.11, 5.12 from <http://www.ccs.neu.edu/~pjd/x600-book-v0903.pdf>

A Flash-based SSD (logical diagram)

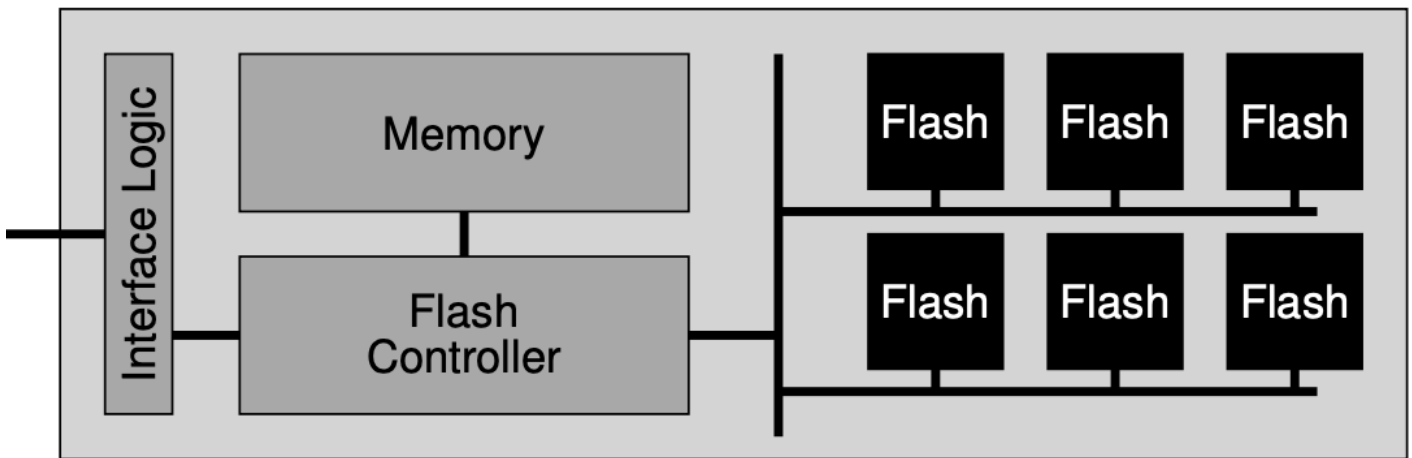


Fig 44.3 from <https://pages.cs.wisc.edu/~remzi/OSTEP/file-ssd.pdf>