

# MAGNETIC STORAGE SYSTEMS

## EE3232 DIGITAL SYSTEMS III CLASS NOTES CHAPTER 12

Department of Electrical Engineering  
University of New Brunswick

© C.P. Diduch

January 5, 2000

### SUMMARY

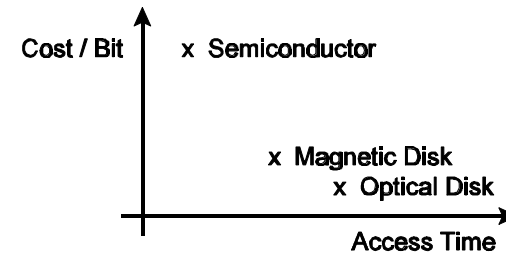
- Objectives.
- Magnetic storage - cost versus access time.
- Magnetic disk storage.
- The disk drive.
- Disk write operation.
- Disk read operation.
- Disk encoding.
- Single density or FM encoding.
- Double density or MFM encoding.
- RLL 2,7 encoding.
- Disk formatting.
- Disk format example.
- The disk operating system.

### OBJECTIVES

- To be able to explain the function of a disk drive and disk controller.
- To be able to explain the details for reading and writing information on a magnetic disk.
- To be able to explain the need for and the distinction between disk encoding and disk formatting.

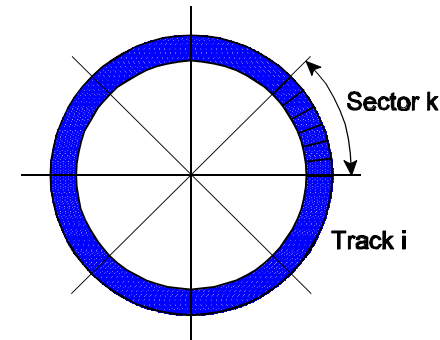
### MAGNETIC STORAGE - COST VERSUS ACCESS TIME

- Magnetic storage systems store large quantities of information at low cost.
- There exists a trade off between cost per bit of storage and access time.



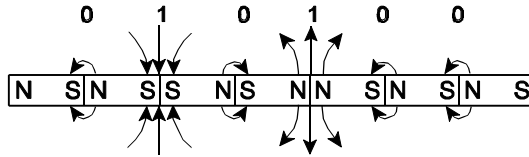
### MAGNETIC DISK STORAGE

- The disk is mylar (floppy) or aluminum (hard) coated with iron oxide.
- Information is saved on the disk,
  - ▶ Bit serial,
  - ▶ In sectors,
  - ▶ Along concentric tracks.



## MAGNETIC DISK STORAGE (CONT'D)

- Each bit is saved as a change in magnetic flux
  - '0' = no change in flux,
  - '1' = a change in flux,



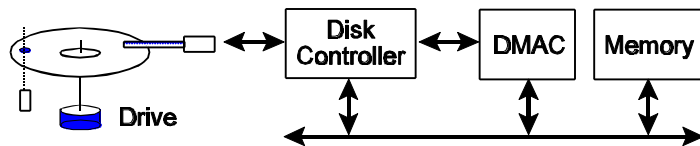
- Most all disks use **soft sectoring** to identify sector boundaries.
- Before a soft sectored disk is used, a special **format** is initially written onto the disk.
- The formatting information is used to identify tracks and the start/end of sectors in subsequent R/W operations.

## THE DISK DRIVE

- The disk drive performs three tasks,

- Spins the disk at constant speed,
- Positions the R/W head over the desired track and
- Transfers data to/from the disk from/to the controller.

- Disk drive components,
  - Track-0 detector,
  - R/W head and stepper motor,
  - Spindle and spindle motor,
  - Disk.

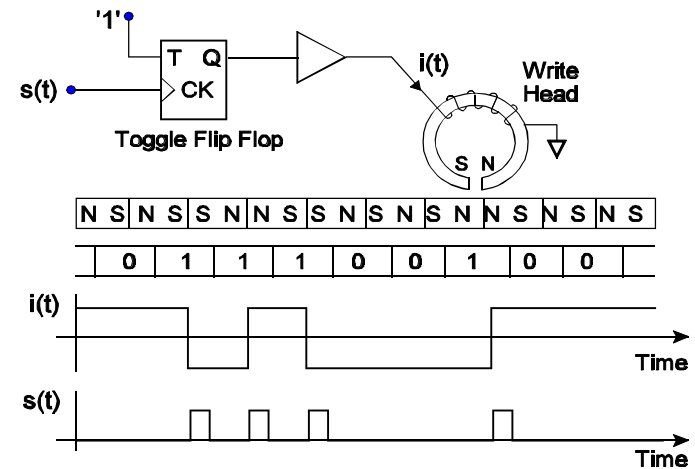


## THE DISK DRIVE (CONT'D)

- How does the disk drive access a sector of information?
  - STEP** the R/W head one track at a time until the desired track is reached. *The disk SEEK TIME = TRACK to TRACK ACCESS TIME x NO of TRACKS.*
  - WAIT** until the R/W stops vibrating. *The duration of the wait is specified by the disk settling time.*
  - LOAD** the read/write head onto the disk. *The duration for this task is specified by the HEAD LOAD TIME.*
  - WAIT** until the desired sector rotates beneath the R/W head. *The maximum duration of the wait is specified by the rotational latency.*
  - TRANSFER** the data (serially) to/from the disk controller.

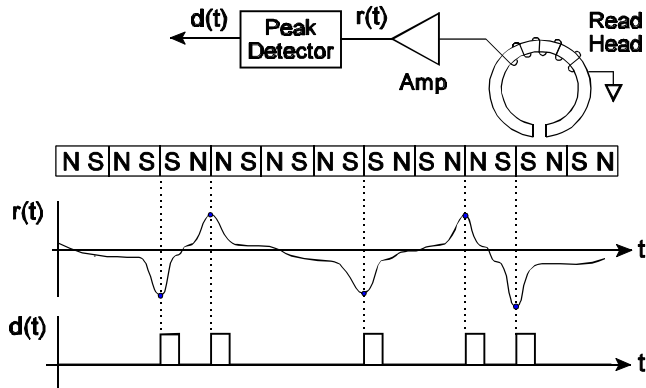
## DISK WRITE OPERATION

- $s(t)$  : serial data written to disk.



## DISK READ OPERATION

- $d(t)$  : serial data read from disk.



## DISK ENCODING

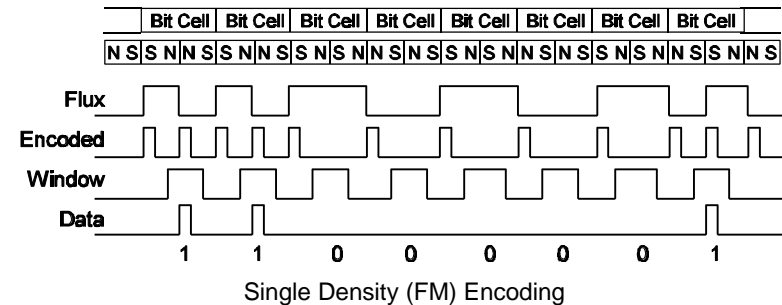
- The speed of the disk spindle motor is not exactly known.
- It is  $\therefore$  not possible to reliably recover the data in each bit cell from flux transitions unless a clock is included.
- In most applications the clock pulses are inserted into the data stream, i.e., **self-clocking** is used.
- The clock is used to adjust the frequency of a PLL (phase lock loop) used for separating data from clock bits.
- The clocking may be classified as,
  1. **Single density** (also called FM encoding)
  2. **Double density** (also called MFM encoding)
  3. **RLL 2, 7**

## SINGLE DENSITY OR FM ENCODING

- Each bit cell begins with a clock pulse.
  - ▶ **Logic '1'**: write a pulse in the middle of the bit cell.

- ▶ **Logic '0'**: no pulse is written.

- There is a maximum of two flux transitions per data bit.



## DOUBLE DENSITY OR MFM ENCODING

- There is a maximum of one flux transition per data bit.
    - ▶ **Logic '1'**: write as a pulse in the middle of a bit cell.
    - ▶ **Logic '0'**: if not preceded by a logic '1', write as a pulse at the beginning of the bit cell (in the clock position), otherwise no pulse is written.
- 
- The diagram illustrates Double Density (MFM) Encoding. It shows a sequence of bit cells: N, S, N, S, N, S, N, S, N, S, N, S, N, S, N, S, N, S. Below the bit cells, four waveforms are shown: Flux, Encoded, Window, and Data. The Flux waveform shows a series of pulses corresponding to the bits. The Encoded waveform shows a series of pulses corresponding to the bits. The Window waveform shows a series of pulses corresponding to the bits. The Data waveform shows a series of pulses corresponding to the bits: 1, 1, 0, 0, 0, 0, 0, 0, 1.
- If the same density of flux transitions are allowed (as FM disks) then twice as much information can be encoded using MFM.
  - The only difference between FM and MFM encoding is the decoding/encoding circuits.

## RLL 2, 7 ENCODING

- Encode the (2 and 4-bit) **data bit sequences** by the **RLL codes**.
- Clocking and data bits are embedded in the RLL codes!

| Data Bit Sequence | RLL 2, 7 Code |
|-------------------|---------------|
| 10                | 1000          |
| 11                | 0100          |
| 000               | 100100        |
| 010               | 001000        |
| 011               | 000100        |
| 0010              | 00001000      |
| 0011              | 00100100      |

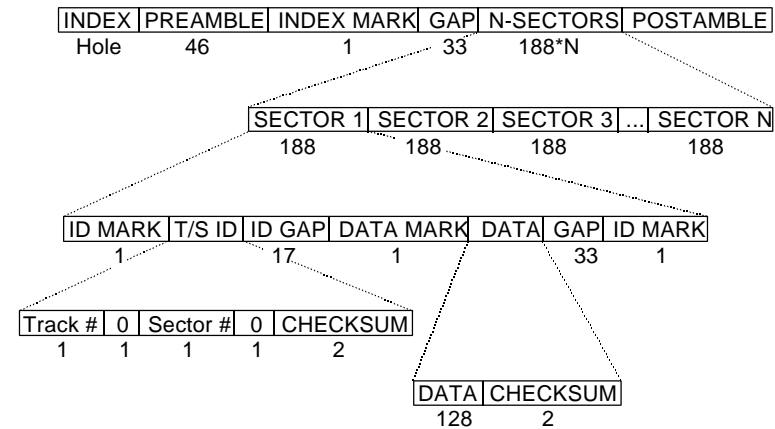
- If data bit sequence = 110000010 then the RLL 2,7 encoded sequence = 0100 100100 00001000.

## DISK FORMATTING

- Before a soft sectored disk can be used it must be **formatted**.
- Formatting initializes the disk by :
  1. Writing track and sector numbers ... at the beginning of each sector,
  2. Writing error detection bits (usually in the form of a Cyclic Redundancy Check word or Checksum) for verification.
  3. Providing gaps of no information to account for the time required to switch from read mode to write mode.
  4. Formatting is software controlled.
  5. Not all computer systems use the same formatting → interchangeability issues.

## DISK FORMAT EXAMPLE

- IBM 3740 floppy disk format.



## THE DISK OPERATING SYSTEM

- Data is saved on a disk in the form of files.
- A file consists of an **integral number of sectors**.
- A disk operating system (DOS) manages files by,
  - ▶ Maintaining a directory of previously defined files, (a copy of the directory is written on the disk),
  - ▶ Deleting files,
  - ▶ Renaming files,
  - ▶ Providing status on remaining disk space.
- The DOS may also include,
  - Editors for creating files,
  - Assemblers / compilers for generating machine code files from ASCII source files,
  - Loaders and debuggers,
  - Print utilities, etc. ...

## **THE DISK OPERATING SYSTEM (CONT'D)**

---

- The DOS consists of a *hierarchy of program modules*,
  - ▶ Low level tasks for positioning the R/W head,
  - ▶ Low level tasks for status checking, ...
  - ▶ Intermediate level tasks for reading or writing a sector of data ...
  - ▶ High level tasks for reading, writing, deleting, copying files ...
  - ▶