Information & Data

Gökçe Aydos

Lecture

Prep

- how does your brain store and process information? How do you know what the number 1 means?
- how does a computer know what 1 means?

Goals

- understand how data is stored in binary
- know data size units, e.g., MB, GiB, and be able to guess their significance
- able to encode data
- understand information compactness & redundancy
- understand data compression

Digital perspective

- computers have a digital perspective to the world
- sees everything in *0*s and *1*s (binary)

Storage vs processing

- in this class we focus on storage

Digital

- having separable states
- noncontinous, discrete
- e.g., *0*, *1*, *-1* (ternary computer)

Encoding vs Decoding

1. keyboard
2. 0101..
3. memory
4. CPU
5. memory
6. monitor
7. pixels
8. brain

Encoding of information

- information -> bits

Bit

- for storing binary information in computers
- potential reason: punched cards existing since 1732

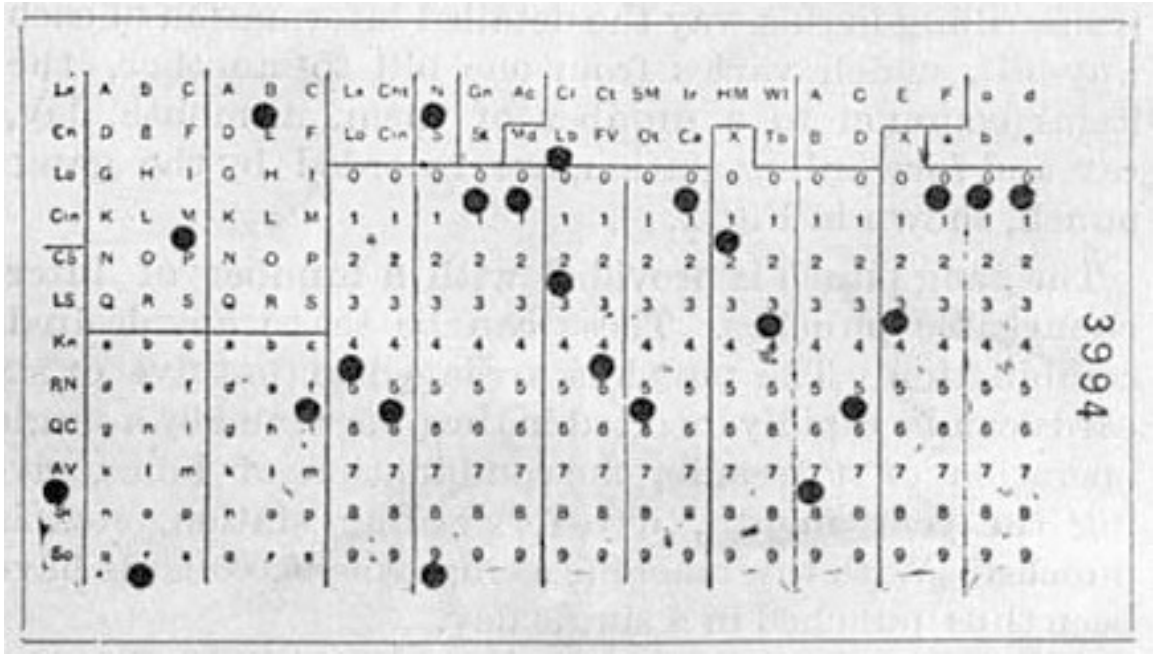


Figure 1

Herman Hollerith [Public domain]

Bit - Easy to store

- two stable states are required
- punched cards
- flip-flops
- two positions of a relay
- two directions of magnetization
- others?

Bit - Binary computers

- bit is used on binary computers
- currently uses smallest amount of resources on electronics
- the industry is based on binary computers

Alternative to bits

- ternary computers using ternary logic
- e.g., optical computing
- 0 off, -1 and 1 for two polarizations of light

Polarization

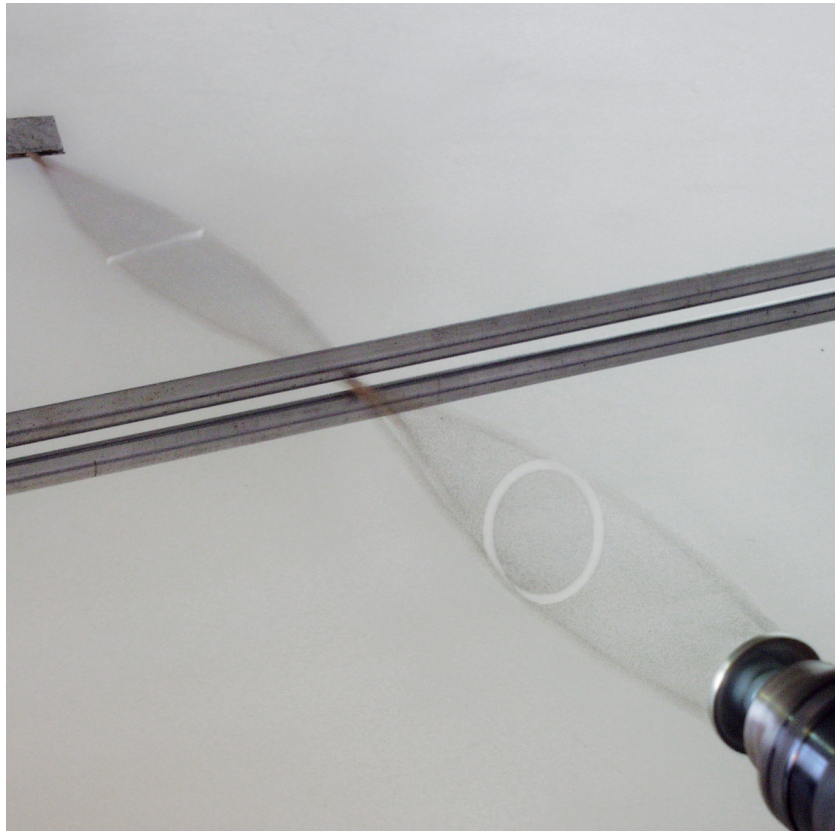


Figure 2: Circular polarization on rubber thread, converted to linear polarization

Example - one bit

0	1
hole	no hole
high voltage	low voltage
magnetized in one direction	magnetized in reverse direction

Nucleotide storage

- adenin
- cytosine
- guanine
- thymine
- how do we store the information about one nucleotide?
- e.g., which single nucleotide did we currently read during genome sequencing?

Bit sequence

- two bits => ability to describe four items
- three bits => eight
- four bits => ?
- also called *bitstring*

Exercise - encoding

- the difficulty rises with each question
  - how many bits do we need to encode a nucleotide triplet, i.e., codon?
  - how many things can we differentiate using a bitstring with a length of exactly `n`?
  - ... a length of *maximum* `n`?

Byte & Kilobyte

- *byte*: 8 bits
- *kilobyte*
- generally 1024 bytes
- should be 1000 bytes

Kibibyte

- solution to 1024 vs 1000 ambiguity
- *kibi* = kilo + binary
- => 1 KiB = 1024 Byte

Exercise - Data size

- which other prefixes do you know?

Binary numbers

- a number consisting of bits
  - 1001
- further material:
  - Saylor Academy — Intro to number systems and binary

Hexadecimal numbers

- compact description of binary numbers
  - addition of A,B,C,D,E
  - A = 10, B= 11, ...

Exercise

- why do we have binary numbers?
- what is the advantage of hexadecimal numbers?
- convert the decimal number 293
  - to a hexadecimal number
  - to a binary number
- convert the hexadecimal number 0xDEAD
  - to a decimal number

Encoding - goals

two different goals:

1. compactness
2. redundancy

Compactness

- minimizing the size of the encoded data for
- saving memory
- saving bandwidth in communications

Compactness - Examples

- data compression
- UTF-8

Exercise - encoding information

- how can we encode natural numbers?
- how can we encode integer numbers?

Exercise - encoding text

- how can we encode the following text? What would be the resulting code?
  - `ACGAATA`
  - `paneer tastes good!`

Example - ASCII Code

symbol	binary	hex
A	01000001	41
B	01000010	42
z	01001010	7a
LF (linefeed)	00001010	0a

Conversion to ASCII - example

- write your name in ASCII Code
- to verify:
- store your name in a text file
- convert it to hexadecimal characters, e.g., the command `hexdump`
- ASCII-table

Encodings for text symbols

- ASCII — 7 bit
- extended ASCII — 8 bit
- Unicode (UTF) — 16 bit
- Universal Coded Character Set (UCS)

Exercise - encoding of images

- a black-white image
- a black-grey-white image
- a color image (hint: RGB)

Exercise - encoding of videos

- a black-white video?
- ...

Redundancy

data can be corrupted

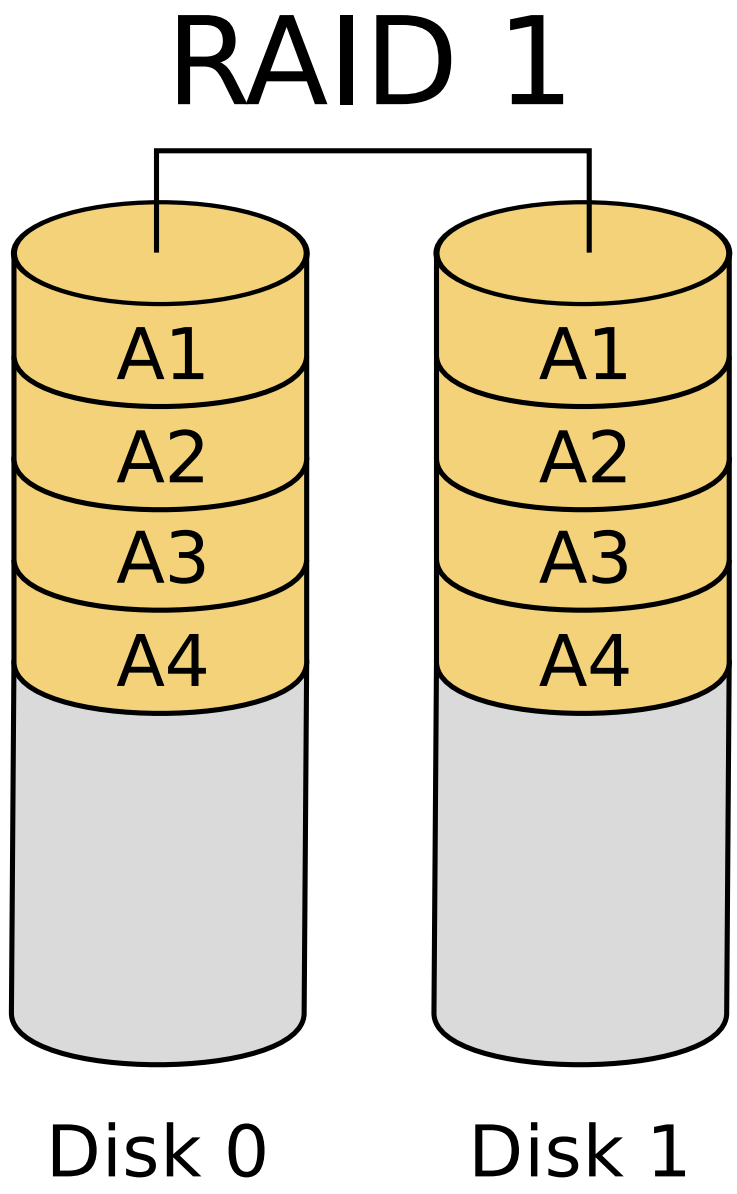
- on transmission
- on storage

Redundancy - Examples

- redundant array of independent disks (RAID)
- check digits on IBAN, DE*91* 1000 0000 0123 4567 89
- writing both your name and matriculation number on the exam sheet

RAID

a method to protect computers against harddisk failures.



RAID - remarks

- used on mostly on servers
- RAID is against hardware failures
- RAID is not a backup solution, it does not help if data is modified or deleted, e.g., by an accident or virus
- you should additionally backup your data

Backups & Security

In your company you are working on a novel idea. How would you deal with your data on your computer?

Backups & Security - Solution

- security: disk encryption (harddrive is encrypted)
- backup: three copies
  - working copy
  - backup
  - an (offsite) backup of your backup
- data organization
  - e.g. big data rather on a server than local
  - probably you will be working with big data on the cloud
- how does THD backup:
  https://intranet.th-deg.de/en/rz/datensicherheit

Exercise - Storing videos

- example:
  - 2h movie
  - 25 frames per second
  - a DVD has 720 x 480 Pixel
  - one pixel needs three bytes
- how much storage do we need?

Solution - Storing videos

- we need about ~186 GB, but DVD max. capacity ~= 17 GB
- how do we fit the video into the DVD?

Data compression

1. lossless
   - e.g., zip, gzip
   - e.g., look for frequent patterns and encode them using shorter bitstrings
   - e.g., only encode differences between frames
2. lossy
   - e.g., jpeg, mp3
   - e.g., use less bits for color encoding

Information vs data

- certain vs uncertain
- certainty achieved through organizing and interpreting data
- information = data put in context and with meaning attached[1]

[1] https://en.wikipedia.org/wiki/Information

Use case: Data Storage using DNA

- fully automated DNA data storage

Summary

- everything on a harddisk is a bitstring
- bit, byte, kB, kiB
- compactness vs redundancy
- lossy vs lossless data compression