Introduction to Computer Engineering (Week-1)

Dr. Tuğberk Kocatekin

Istanbul Arel University

Introduction

This course aims to prepare you for upcoming courses. We will introduce several topics and start programming with Python.

- Data storage & manipulation
- Operating Systems
- Networking & Internet
- Algorithms
- Programming Languages
- Software Engineering
- Database Systems
- etc.

Career Paths

- Computer Engineering ≠ Software Engineering
- Several Paths: Business Analyst, Product Management, DevOps, Software Engineer, Testing engineer, etc..

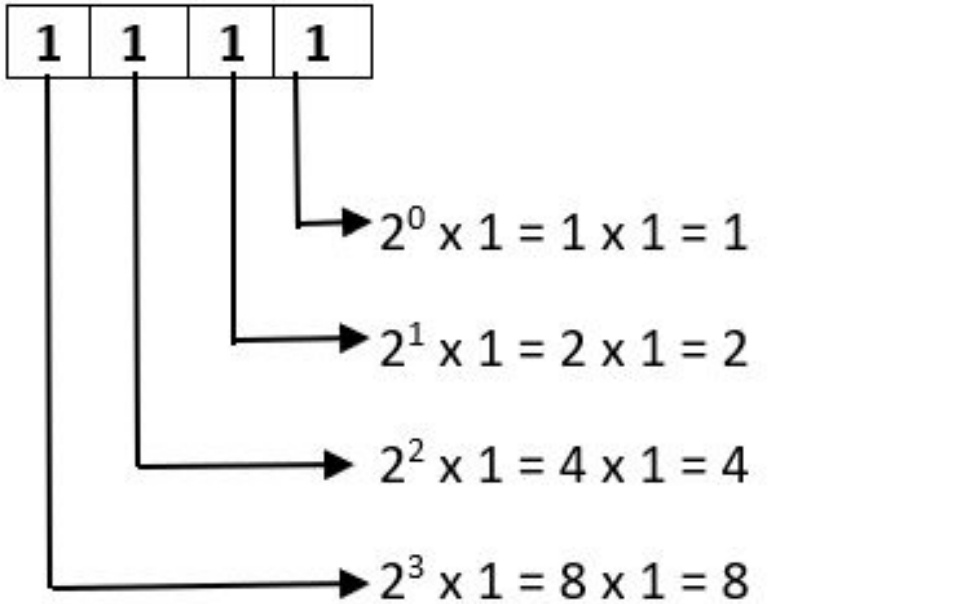
For many software jobs it may seem you don’t need a thorough education, but you do. Remember that you are going to be engineers, not programmers. You may choose to do programming but your education will be more.

Binary numbers

- Base-2 numeral system (*binary numeral system*) 0 and 1.
- In computer systems, these are called **bits** (binary digit).
- In programming, they are usually implemented as data type **boolean**.
- In electronics, it represents voltage (*high or low*).

- A byte consists of 8 bits.
- Everything is represented as bits in a computer.
- We use Base-10 (decimal) number system where digits are from 0 to 9.
- A fun game you can play with: (Click here)

Binary conversion



Resultant decimal number= 1+2+4+8 = **15**

Hexadecimal system

- Base-16 system
- An easy way for streams of bits.

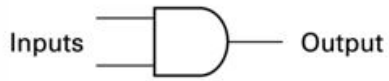
Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Logic Gates

Here are the *Truth Tables* of the well-known logic gates.

AND Gate

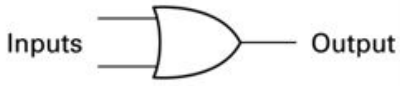
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



When all inputs are 1, the output is 1. It can be represented as `&&` or `and` in programming languages.

OR Gate

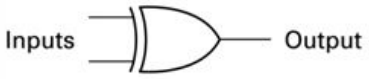
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



When any input is 1, the output is 1. It is represented as `||` or `or` in programming languages.

XOR Gate

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



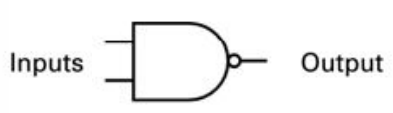
If the inputs are different, the output is 1. Widely used in cryptography. Represented as `^`.

Logic Gates (cont.)

There is also a NOT gate which gave way to several more gates such as NOR, XNOR, and NAND.

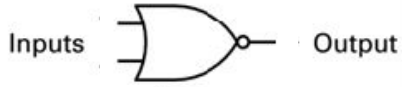
NAND Gate

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



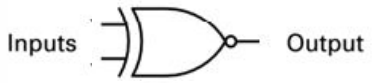
NOR Gate

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



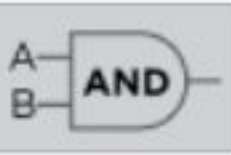
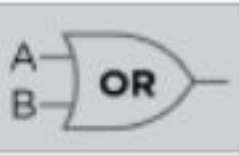
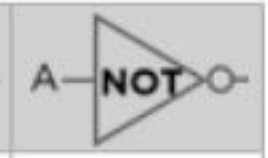
XNOR Gate

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



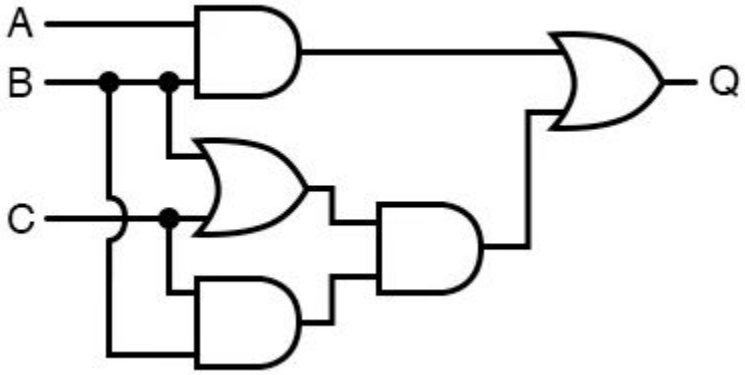
Examples for Logic Circuits

Boolean logic is created by mathematician George Boole. Remember that it is also a **data type**. The logic is built upon two Boolean values: **true** and **false**. Here below is the truth table which combines three fundamental operators (*AND, OR, NOT*) and using **true** or **false** instead of 1 and 0.

A	B			
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

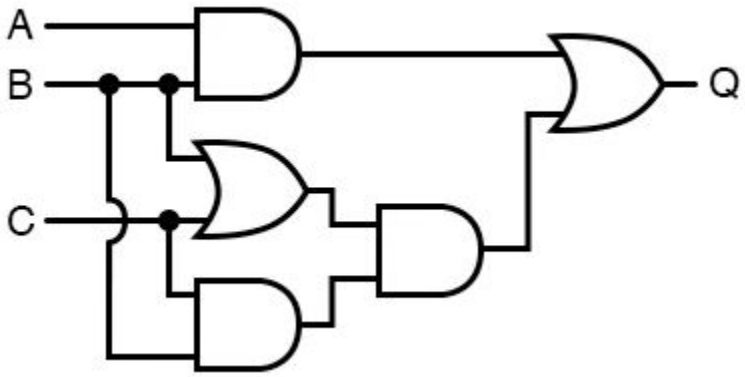
Example

Let us create the truth table for the logic circuit below:



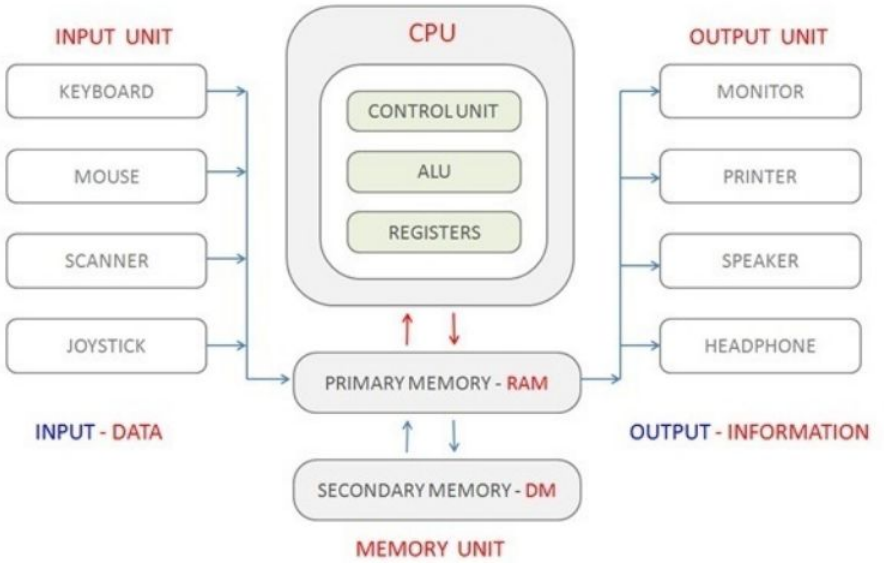
Example

Let us create the truth table for the logic circuit below:



A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

What is a computer?



Example image of a computer system[1]

A computer processes input data as per given instructions to generate desired output. As seen on the figure, a basic computer has CPU, Memory, I/O and Storage. All these work together to deliver the desired output.

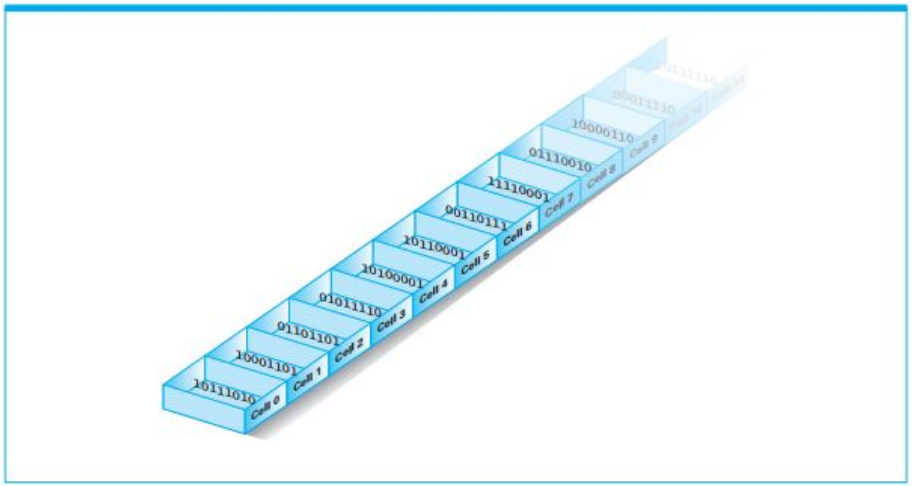
Main memory

- Main memory is organized in units called **cells**.
- Typical cell size is 8 bits (**byte**).
- To identify cells, they are given a name: **address**

- Using read & write operations, other circuits can get and write data to memory using addresses.
- These cells can be accessed independently. That is why it is called **RAM: Random Access Memory**.

Address

- Cell size is 8bits: 00000001
- While programming you generally refer to addresses using Hexadecimal representation. (**0x** in the front means it is hex)
- Every hex represents 4 bits.



Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

32-bit vs 64-bit

- CPU registers stores memory addresses. That is how the processor accesses RAM. One bit in the register can reference to an individual byte in memory.
- You can’t have more addresses than memory bytes.
- 32-bit means the memory address can be 32bit (8 hex rep) (0x0A030101)
- E.g. If CPU can’t store 10; how can it reach 10th address?

0x7FFE4A71	1010 0001
0x7FFE4A72	1101 1010
0x7FFE4A73	0101 0100
0x7FFE4A74	1001 1000
0x7FFE4A75	1010 0101
0x7FFE4A76	0101 0101
0x7FFE4A77	1110 0111

1000 vs 1024

- Kilo, mega, etc. represents 1000.
- Kilometer = 1000 meter
- Kilobyte = 1000 byte

- However, it is actually not 1000 but 1024 (2 ^ 10)
- They used kilo because 1000 is close to 1024.
- In 1990, a new standard arose using *kibi, mebi, gibi, tebi. 100KiB*
- However, their usage is not widely adopted.

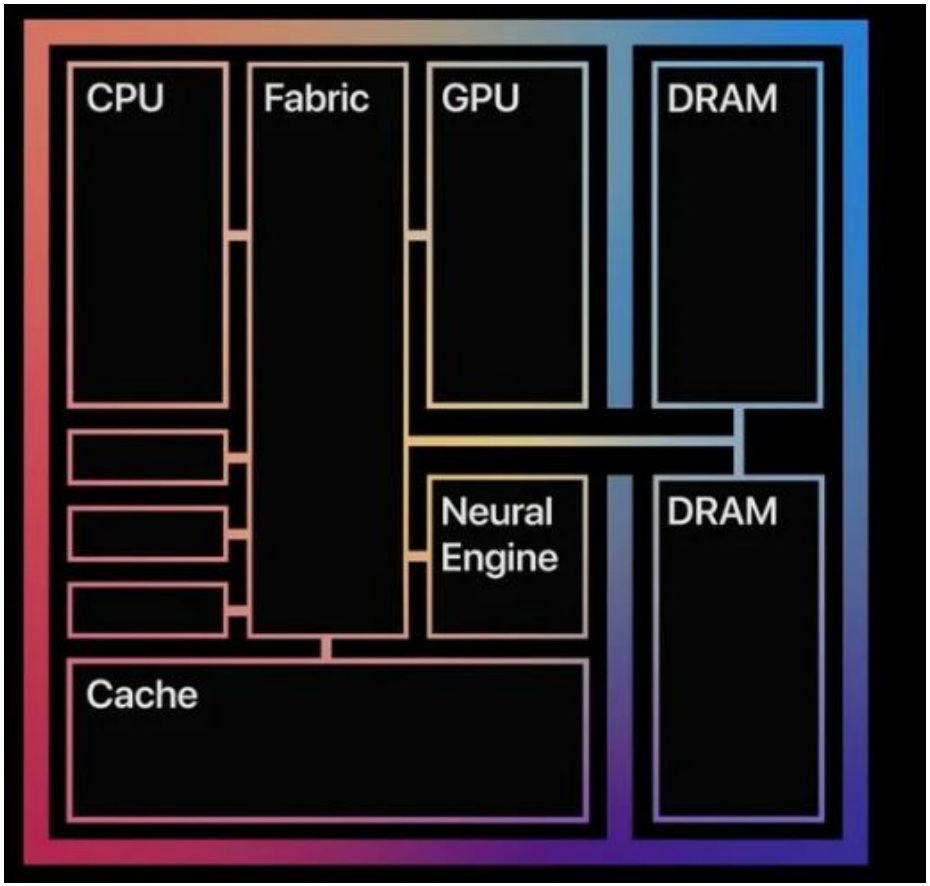
- In computer science, when you say kilo etc you actually mean 1024.

Microprocessors

- Every CPU is a microprocessor, but not every microprocessor is a CPU. A microprocessor can also contain a GPU.
- CPU is single. Microprocessors are IC (integrated circuit).
- Can perform arithmetic and logic operations (ALU).

CPU vs M1

- M1 is a SoC (system on a chip).



Introduction to Python

- An interpreted and object oriented language
- Can do many tasks:
  - Turtle: Graphics
  - Command Line Applications (CLI) using *Click, Typer, Rich, etc.*
  - GUI Applications (Graphical User Interface) using *tkinter, PyQt, etc*
  - Game development using *PyGame, RenPy, etc.*
  - Data Science / Mining using *Numpy, Scikit-Learn, Pandas, etc.*
  - Web Applications using *Django, Flask, FastAPI, etc.*
  - Machine Learning using *NLTK, scikit-learn, TensorFlow, etc.*
  - Web Scraping using *requests, Beautiful Soup, etc.*
  - Much more...

Why Python?

- Easy learning curve
- Start coding easily
- Comes pre-installed in GNU/Linux
- Uses indentation instead of “;”
- Uses a package manager: **pip**

Java

```java
public class Main {
  public static void main(String[] args) {
      System.out.println("hello world");
  }
}
```

Python

```python
print("hello world");
```

Python

- If, elseif, else
- For, while
- Lists, tuples, dict
- String, int, float, boolean
- ...

References

[1]: learncomputerscienceonline.com

Data representation & detection & correction

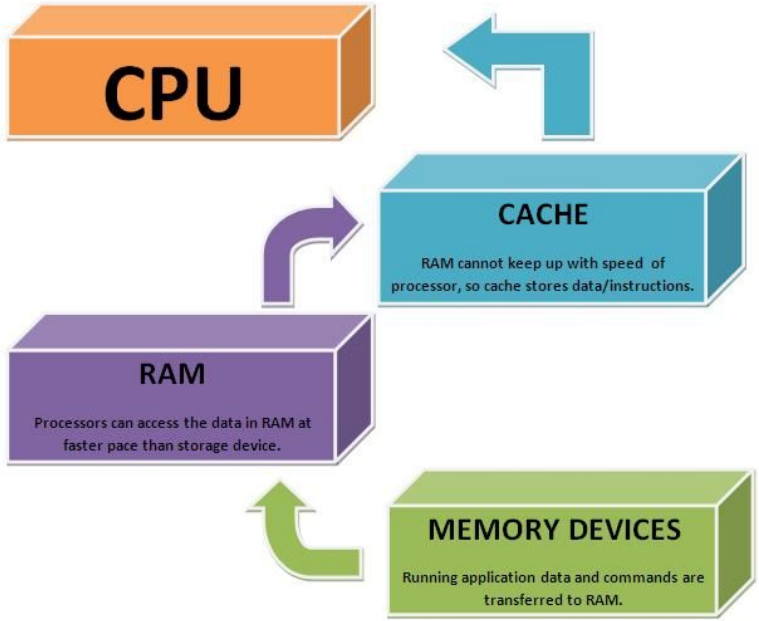
Dr. Tuğberk Kocatekin
Istanbul Arel University

RAM (cont.)

- RAM size vs. speed
- Two types:
  - SRAM (CPU has it)
  - DRAM (The one we use)

Cpu Cache

- 2 types of RAM: SRAM & DRAM
- SRAM is faster
- CPU has faster memory units in itself, they are SRAM.



Cpu Cache (L1, L2, L3)

- L1: Fastest and has the data CPU is most likely to need.
  - Instruction cache
  - Data cache
- L2: Slower than L1 but bigger. Still faster than RAM. (L1 almost x100, L2 x25)
- L3: Largest but slowest.

ROM

- Read Only Memory.
- There are several types.
- Traditionally, you can only write information in the manufacturing phase. It is only for reading data.

Data representation (text)

- Every letter is represented as a bit.
- **ANSI**: American National Standards Institute
- **ASCII:** American Standard Code for Information Interchange
  - 7 bits to represent uppercase and lowercase letters of the English alphabet
- Hello: 01001000 01100101 01101100 01101100 01101111
- **ISO**: International Organization for Standardization
  - Developed number of extensions to ASCII for other letters and symbols

Data representation (text)

- ASCII does not cover many languages. A document can have symbols and letters from different languages.
- Unicode is the answer!
- UTF-8: Unicode Transformation Format 8-bit
    - ASCII can still be represented with 8 bits, additional chars can be rep with 16bits. UTF-8 calso uses 24, 32 bit patterns.

Data representation (images)

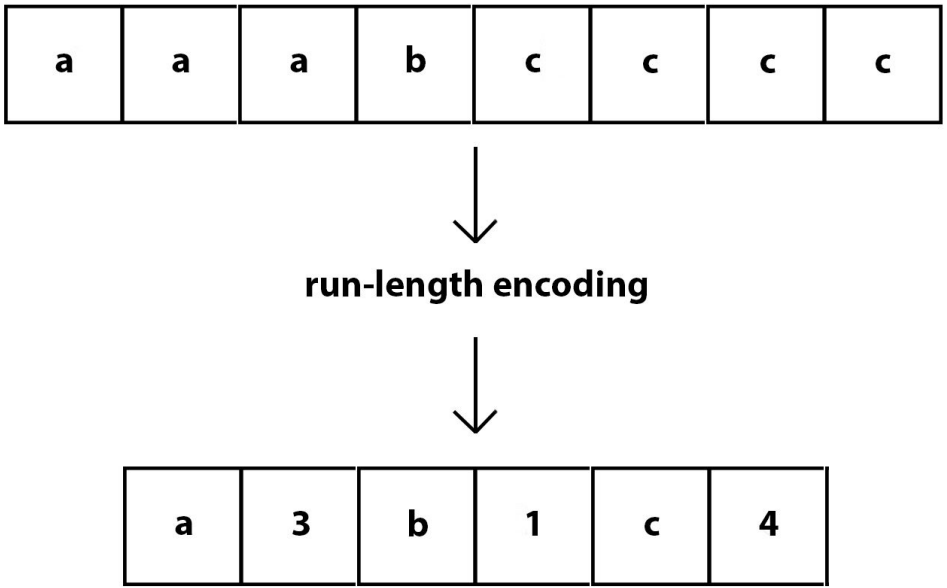
- Image is a collection of dots (**pixel**).
- An image is represented as a collection of encoded pixels. → **bit map**!
- B&W image can be represented with 0, 1 (or more depending on the quality of grayness)
- Color images can be represented where each pixel is represented as a byte of three components: **RGB** (Red-Green-Blue)

Data compression

- Reduce the size of data without corrupting the information it has.
  - Loseless: Lose no information in the compression process
  - Lossy: Lose information, more compression.
- If minor errors are tolerable, lossy can be beneficial such as images and audio.
- Some simple lossless compression algorithms:
  - Run-length encoding
  - Frequency-dependent encoding
  - …

Run-length encoding

- Replace sequences of identical data elements.



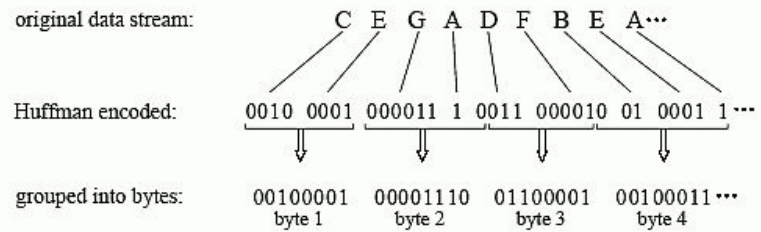
Frequency-dependent encoding

- Most frequency-dependent encoding are called Huffman codes. David Huffman discovered an algorithm that is commonly used for developing FDE.
- Example:
  - In the english language some letters are used more frequently compared to others. Therefore you can use short bit patterns to represent frequently used. Therefore the output will be shorter compared to with uniform-length codes.

FIGURE 27-3
Huffman encoding. The encoding table assigns each of the seven letters used in this example a variable length binary code, based on its probability of occurrence. The original data stream composed of these 7 characters is translated by this table into the Huffman encoded data. Since each of the Huffman codes is a different length, the binary data need to be regrouped into standard 8 bit bytes for storage and transmission.

Example Encoding Table

letter	probability	Huffman code
A	.154	1
B	.110	01
C	.072	0010
D	.063	0011
E	.059	0001
F	.015	000010
G	.011	000011

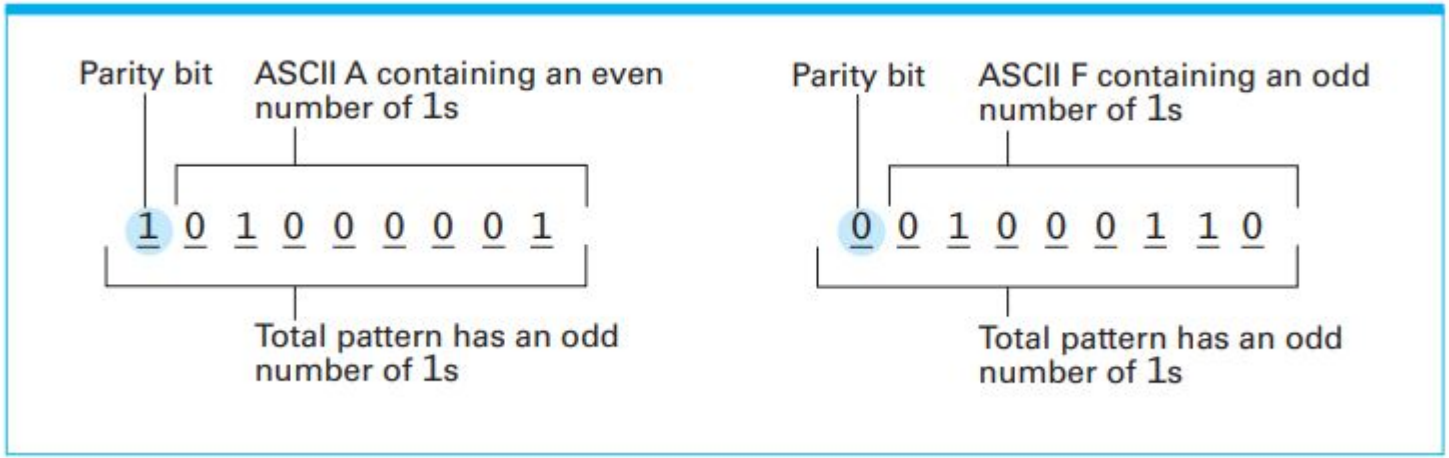


Compression

- GIF (Graphic Interchange Format)
- Jpeg (Joint Photographic Experts Group)
- Mpeg (Motion Picture Experts Group)
- MP3
- ..
- These are all compression formats.

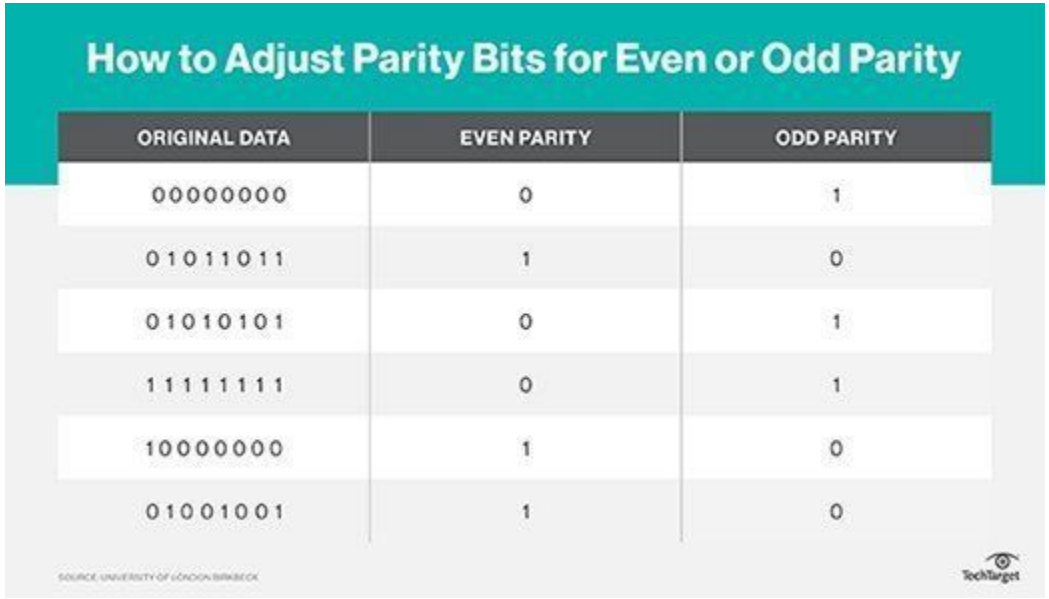
Error correction

- While data is transferred, data can get corrupted.
- Parity bit:
  - A simple method of detecting errors. (Odd parity & even parity)
- You add 1 or 0 if the resulting pattern has an odd number of 1s. (odd parity)



Parity bit

- Adds an additional bit
- Can only detect odd (even) bit errors.



How to Adjust Parity Bits for Even or Odd Parity

ORIGINAL DATA	EVEN PARITY	ODD PARITY
00000000	0	1
01011011	1	0
01010101	0	1
11111111	0	1
10000000	1	0
01001001	1	0

SOURCE: UNIVERSITY OF LONDON BIRKBECK

TechTarget

Checkbyte

- Parity bit is limited
- **Checkbyte** has a collection of parity bits.
- Lead to well-known error detection schemes called **checksums** and **CRC**’s.