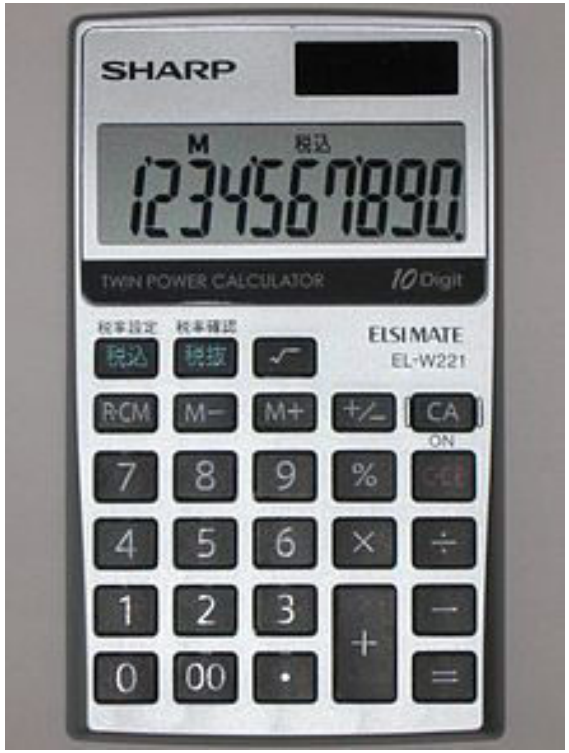
04

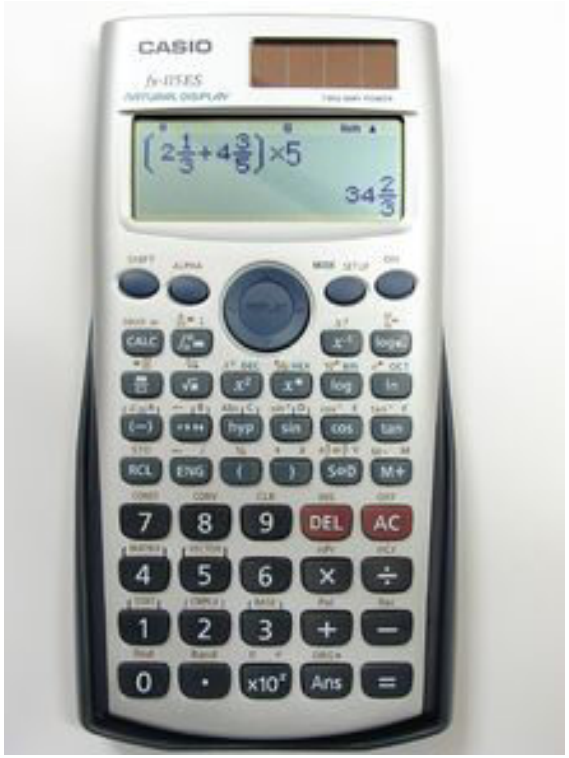
Computer System

Pengantar Teknik Informatika (HUG1M2)

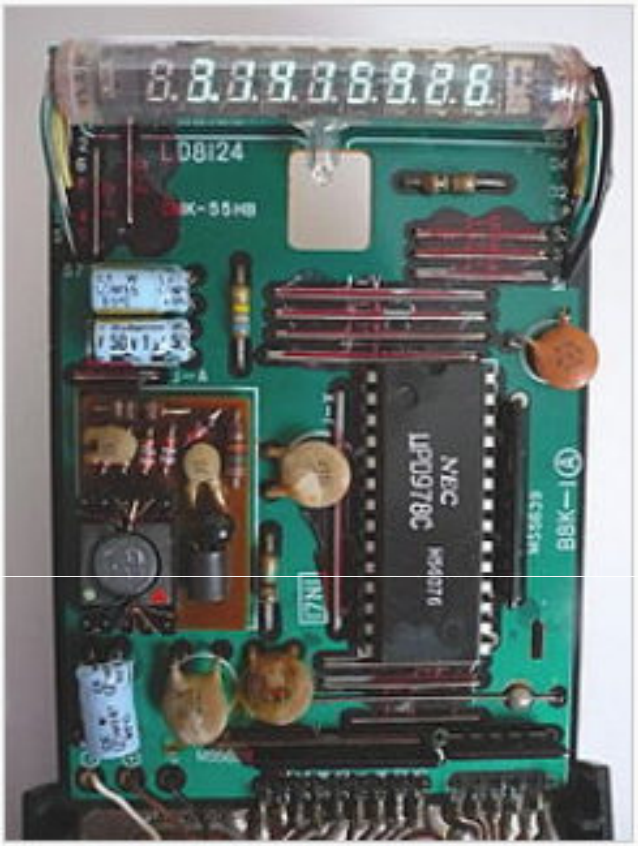
20131

Electronic Calculator

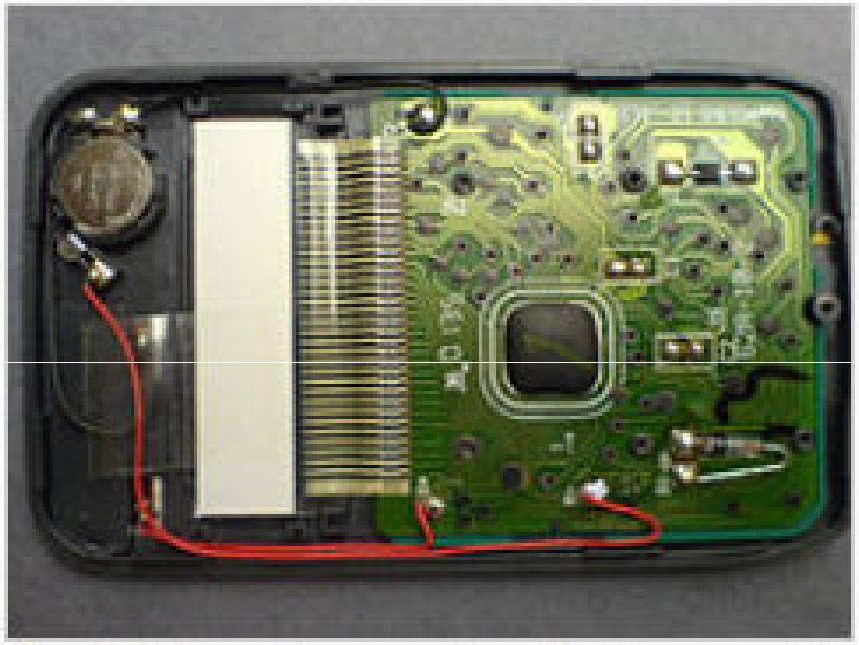




http://en.wikipedia.org/wiki/Calculator



The interior of a Casio fx-20 scientific calculator from the mid-1970s, using a VFD. The processor integrated circuit (IC) is made by NEC. Discrete electronic components like capacitors and resistors and the IC are mounted on a printed circuit board (PCB)



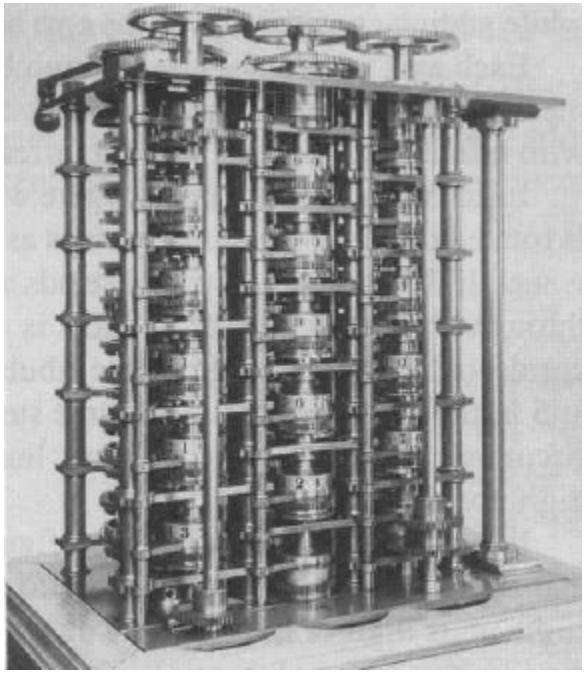
The interior of a newer (ca. 2000) pocket calculator. The processor chip (not visible) in the centre is covered with dark epoxy

http://en.wikipedia.org/wiki/Calculator

Difference Engine (1820)

- Charles Babbage 

- proofreading a set of tables prepared for astronomical calculations . In a moment of exasperation with the errors they found, Babbage remarked, **"I wish to God these calculations had been executed by steam."**



Analytical Engine (1834…1837)

- Difference Engine: each storage axis is also an adder.
- Analytical Engine: there is a separate "**store**" for numbers and a "**mill**," or arithmetic unit, where calculations are made
  - Never built

In its logical design the machine was essentially modern, anticipating the first completed general-purpose computers by about 100 years. (wikipedia)

Trial model of a part of the Analytical Engine, built by Babbage[1], as displayed at the Science Museum (London)

- Human ‘‘computers’’ at work at North American Aviation, Los Angeles, in the early 1950s

Ten years later…

- A pair of IBM 7090 computers assist in the design and testing of the rocket engines that will later take men to the Moon and back
- of the four men visible here, two are employees of IBM, not NAV

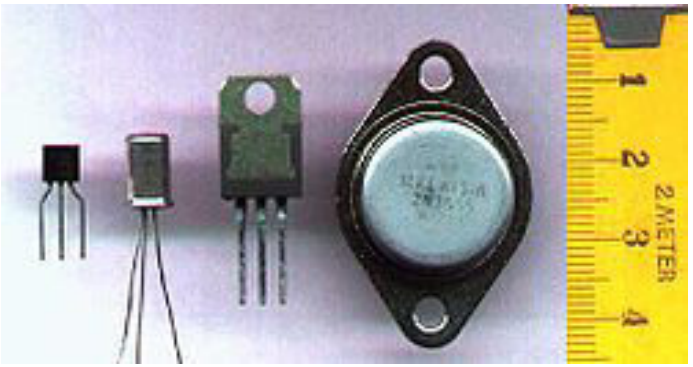
1940s – 1990s

- the computer's transformation in the late 1940s from a specialized instrument for science to **a commercial product**,
- the emergence of **small systems** in the late 1960s
- the advent of **personal computing** in the 1970s
- the spread of **networking** after 1985.

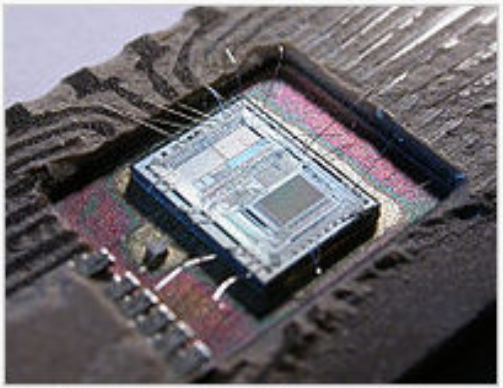
3 generations

- Vacuum tubes
- Transistors
- Integrated Circuits

Tubes from a 1950s computer.



This 8749 Microcontroller stores its program in internal EPROM



The integrated circuit from an Intel 8742, an 8-bit microcontroller that includes a CPU running at 12 MHz, 128 bytes of RAM, 2048 bytes of EPROM, and I/O in the same chip.

AMD Geode is an x86 compatible system-on-a-chip

Common threads that have persisted

- internal design of the computer
- Despite the changes in implementation from vacuum tubes to integrated circuits, the flow of information within a computer, at one level at least, has not changed.
- This design is known as the ‘‘**von Neumann Architecture**,’’ after John von Neumann (1903–1957)

von Neumann Architecture

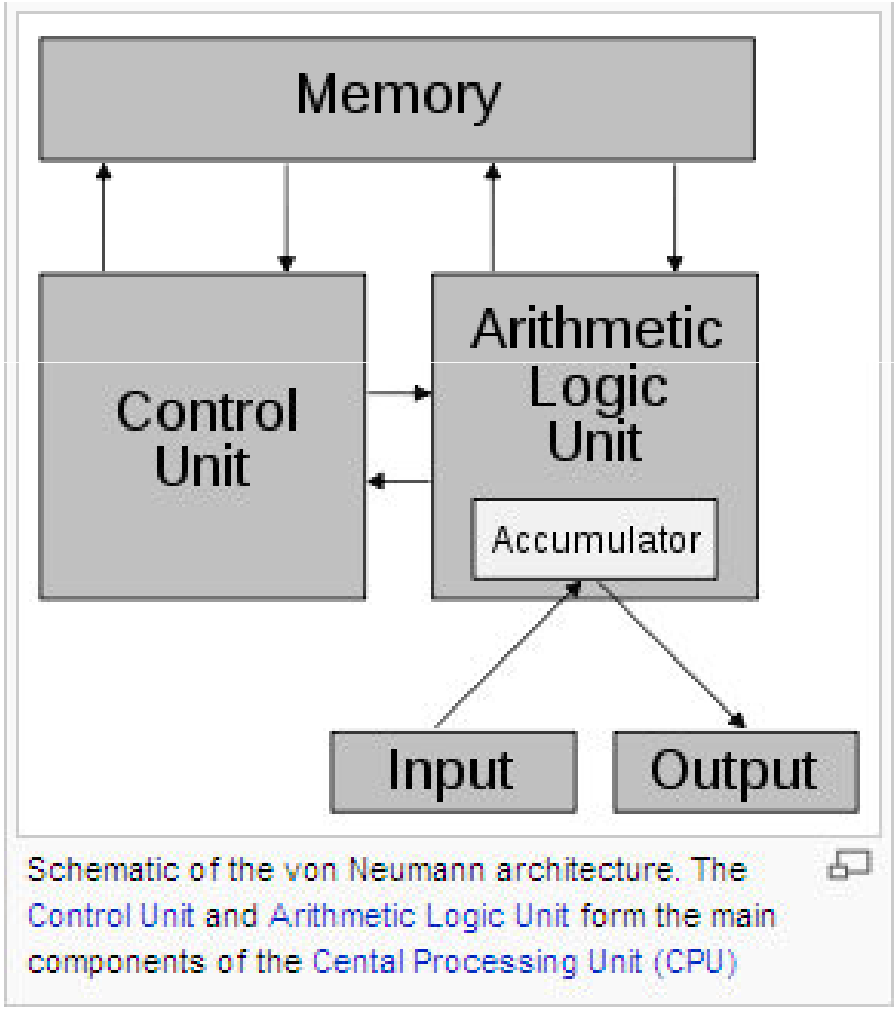
- ‘‘First Draft of a Report on the EDVAC,’’ by John von Neumann, dated June 30, 1945, is often cited as the founding document of modern computing

- **internal storage** of programs
- the units that **process** information are separate from those that store it.
- typically there is only a **single channel** between these two units

Von Neumann (cont'd)

- instructions and data are stored in the **same memory device**, from which any datum can be retrieved as quickly as any other
  - one may treat a coded instruction as a piece of data and perform an operation on it, thus changing it into another instruction

Von Neumann (cont'd)



Schematic of the von Neumann architecture. The Control Unit and Arithmetic Logic Unit form the main components of the Cental Processing Unit (CPU)

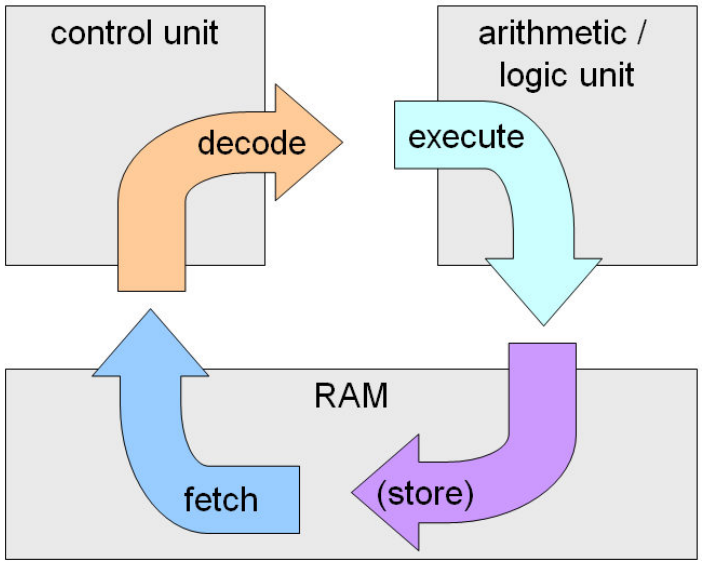
von Neumann: first implementation

- UNIVAC main store could hold up to 1,000 ‘‘words,’’ which could either be
  - numbers (11 digits plus sign),
  - characters (12 characters per word), or
  - instructions (6 characters per instruction; 2 in each word)

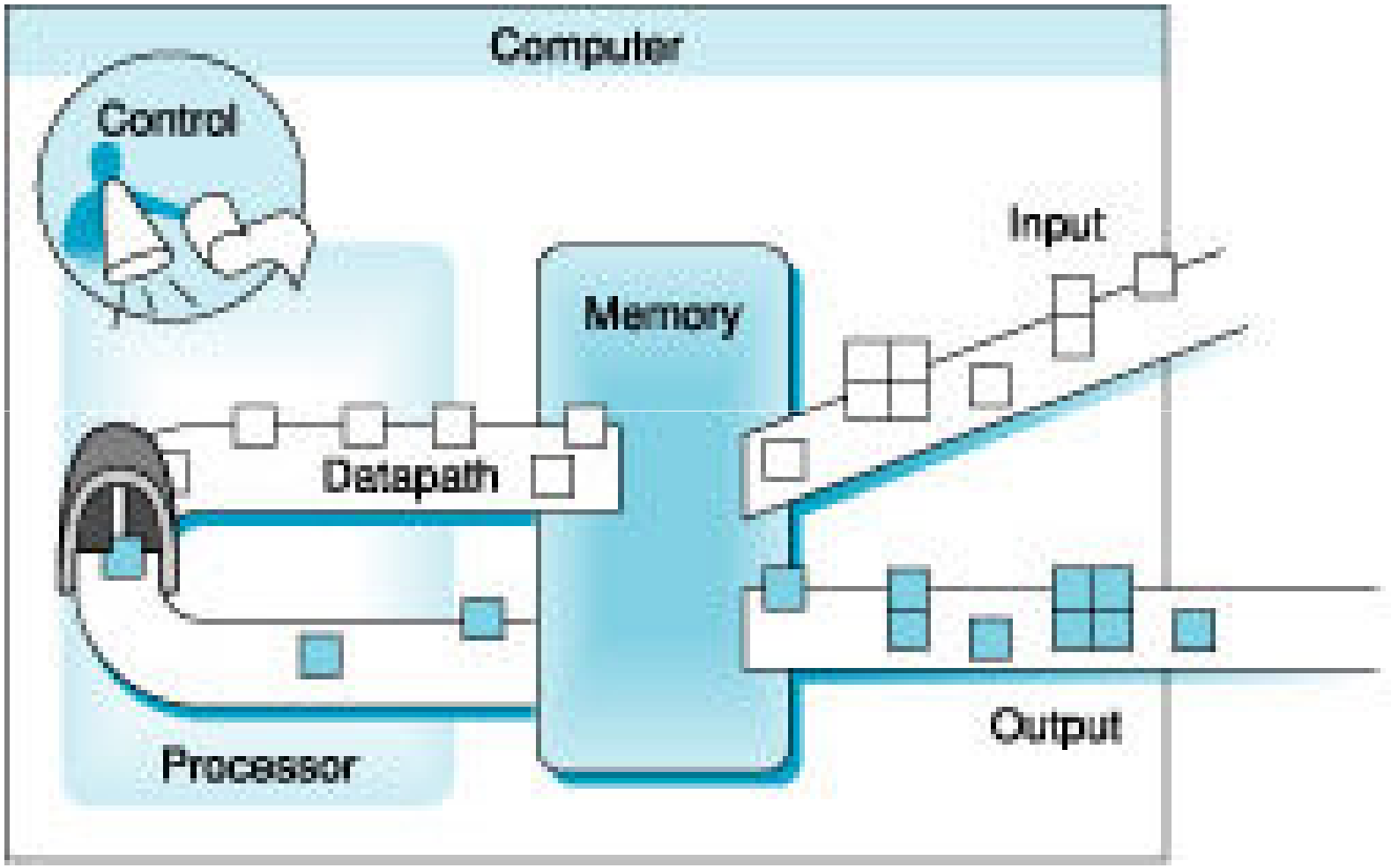
von Neumann: the basic cycle

- transfer an instruction from the store to the processor (**fetch**),
- **decode** that instruction, and
- **execute** it, using data retrieved from that same store or already present in the processor

Except when explicit **branch** instructions are encountered, the flow through the instructions stored in the memory was **sequential** and linear

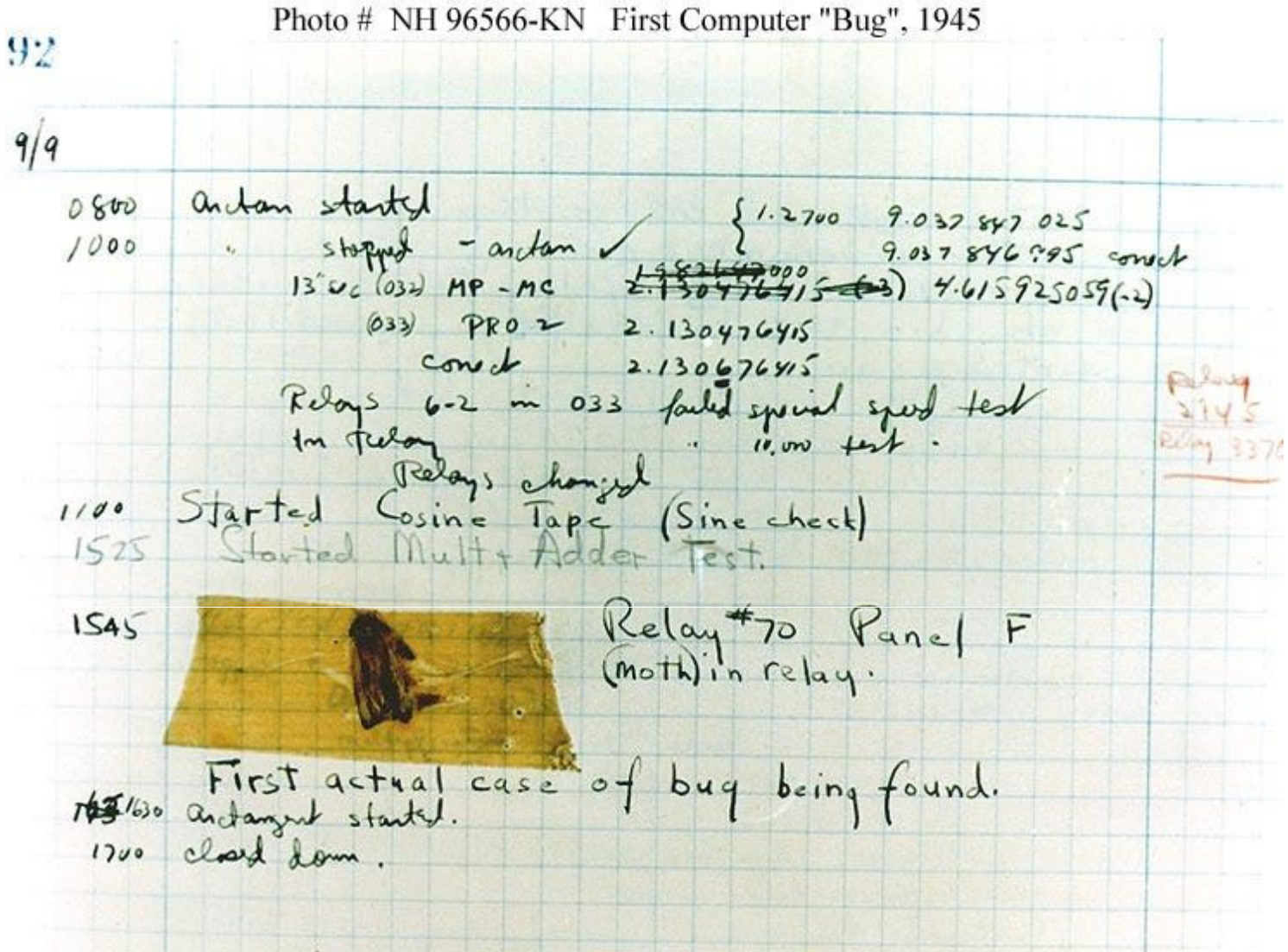


Major Components of a Computer



Fungsi Komputer

- Fungsi apa saja yang terdapat di dalam komputer ?
  - *Data processing*
    - Penjumlahan, pengurangan, konversi terhadap data, dll
  - *Data storage*
    - Penyimpanan data hasil pemrosesan
    - Pengambilan data yang telah disimpan
  - *Data movement* (I/O, komunikasi data, ...)
    - Pergerakan data internal (di dalam komputer)
    - Pergerakan data eksternal (komputer dengan peralatan lain)
  - *Control*
    - Pengendalian ke-3 proses di atas



- *1945 - Working on a prototype of the Mark II, in the summer Grace Murray Hopper finds the first computer "bug," a moth that had caused a relay failure*

Representasi Informasi

Non positional number system

```
  XXVII
    III
-------- x
```

Positional number system

$$d_n d_{n-1} d_{n-2} \ldots d_3 d_2 d_1 d_0 =$$

$$d_n r^n + d_{n-1} r^{n-1} + d_{n-2} r^{n-2} \ldots d_3 r^3 + d_2 r^2 + d_1 r^1 + d_0 r^0$$

d = nilai bilangan;
r = radix (basis bilangan) = jumlah simbol maksimum
n = posisi bilangan, LSB = posisi ke-0

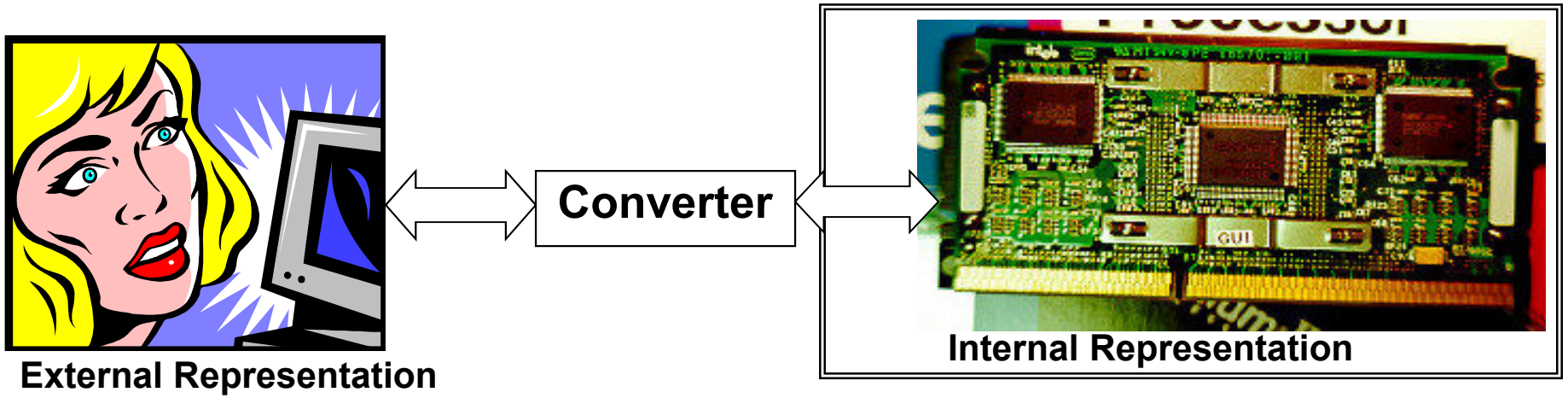
Mana saja yang tidak benar?

- 1 + 1 = 10
- 6 + 7 = 15
- 10 + 24 = 34
- 18 + 18 = 30

Masih ada yang tidak benar?

- $1_{(2)} + 1_{(2)} = 10_{(2)}$
- $6_{(8)} + 7_{(8)} = 15_{(8)}$
- $10_{(10)} + 24_{(10)} = 34_{(10)}$
- $18_{(16)} + 18_{(16)} = 30_{(16)}$

Representasi Informasi (1)



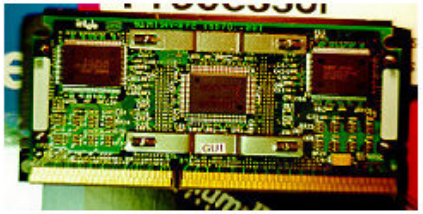
Representasi Eksternal adalah suatu cara untuk merepresentasikan dan memanipulasi informasi oleh *programmer* dengan suatu bahasa pemrograman atau notasi bahasa perintah lainnya –> Agar nyaman bagi programmer (user).

Representasi Internal adalah suatu cara untuk menyimpan dan memanipulasi informasi secara aktual di dalam sistem komputer –> Agar mudah dalam membangun perangkat keras.

Informasi ≈ program & data ≈ deretan bit

→ akses/manipulasi terhadap informasi ≈ akses/operasi (arithmetic/logic) terhadap deretan bit

Representasi Informasi (2)



$-\infty \ldots \infty$	*Finite precision number*
Decimal : $X_{(10)}$	Binary : $X_{(2)}$

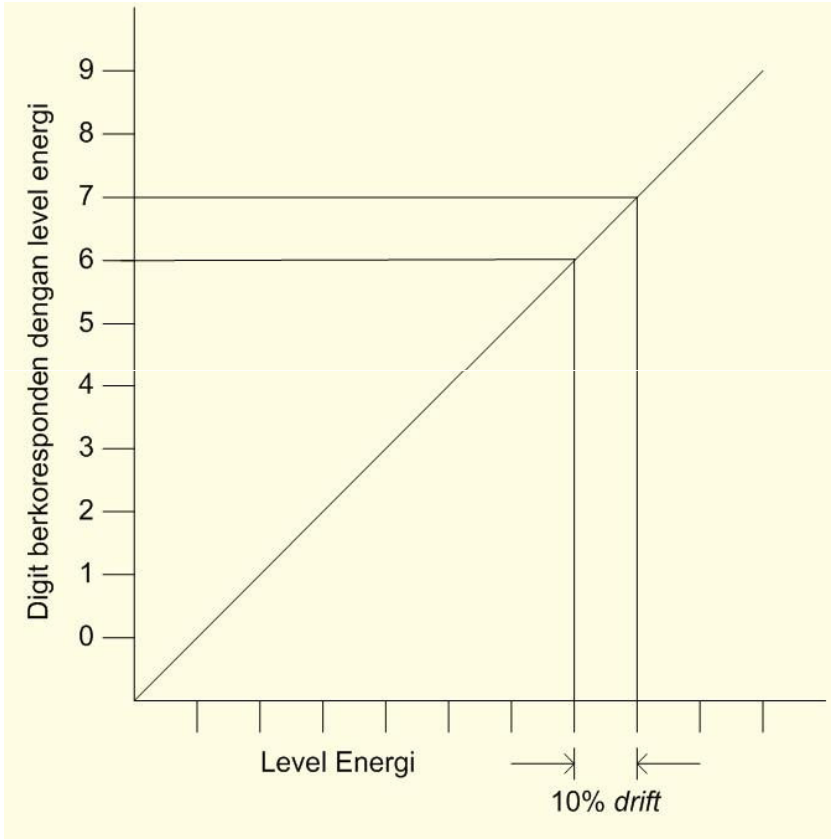
- Bilangan berpresisi terbatas berpeluang memunculkan ‘kesalahan’ (dari segi matematika klasik), tetapi bisa menjadi ‘kebenaran’ sebagai konsekuensi logis dari keterbatasan mesin tersebut
- Kesalahan yang dapat terjadi:
  - *overflow error*
  - *underflow error*
  - *unrepresentable*

Bit dan Byte

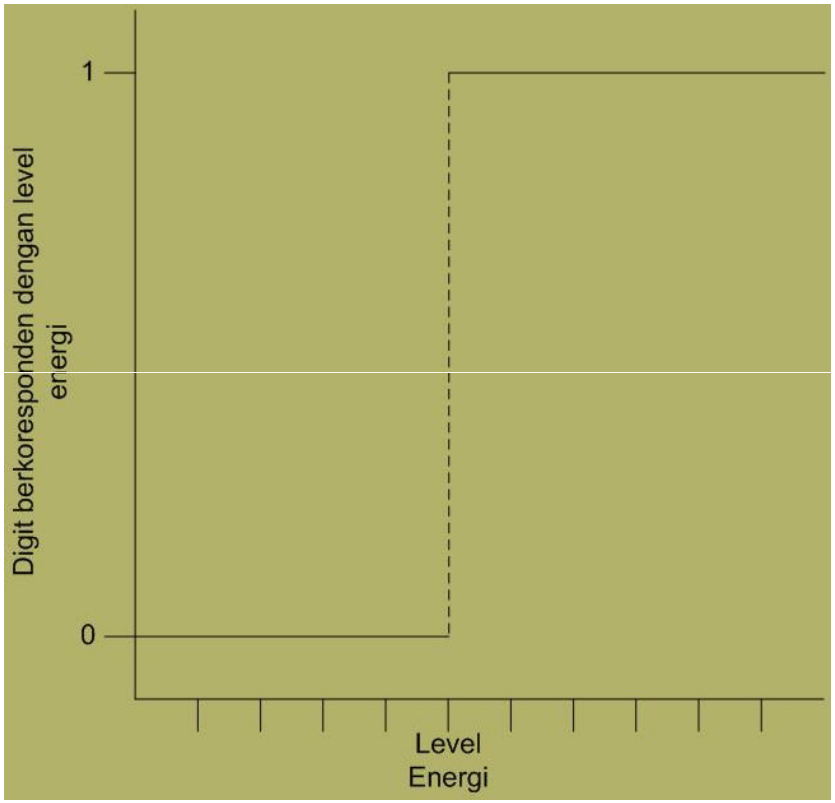
- Apa bedanya antara bit dan byte ?
- 1 byte = 8 bit (binary digit)
  - Range Binary: 00000000_2 - 11111111_2
  - Range Decimal: 0_{10} - 255_{10}
  - Range Hexadecimal: 00_{16} - FF_{16}
    - representasi bilangan basis 16
    - Menggunakan karakter ‘0’ - ‘9’ dan ‘A’ - ‘F’
  - Range Octal: ... - ...
    - 000_8 – 377_8

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Bilangan Desimal & Biner



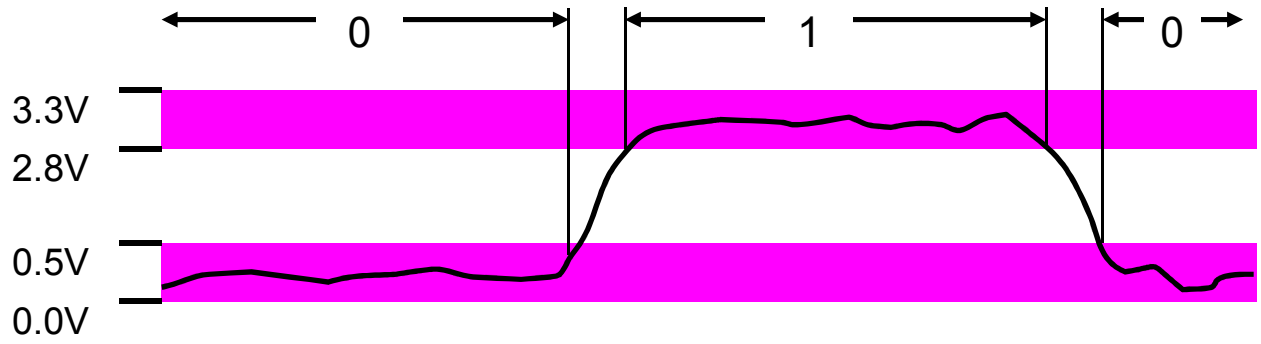
Desimal



Biner

Bilangan Biner

- Representasi bilangan basis 2
  - Representasi $15213_{10} = 11101101101101_2$
  - Representasi $1{,}20_{10} = 1{,}00110011001100110011[0011]\ldots_2$
  - Representasi $1{,}5213 \text{ X } 10^4 = 1{,}1101101101101_2 \text{ X } 2^{13}$
- Implementasi Elektronik
  - Mudah untuk disimpan sebagai elemen yang bistable (hanya ada 2 nilai yang berbeda jauh)
  - Lebih handal pada *wire* yang ber-noise dan inaccurate
  - Mudah diimplementasikan pada fungsi logika digital



Jenis-Jenis Bilangan Biner

- Bilangan bulat biner tak bertanda (unsigned integer)
- Bilangan bulat biner bertanda (signed integer)
  - Sign/magnitude
  - Komplemen 2 (radix complement)
  - Komplemen 1 (diminished radix complement)
  - Binary Coded Decimal (BCD)
- Bilangan pecahan biner (floating point)

Konversi Bilangan: Desimal → Biner

Desimal ke biner

$43_{10} = \ldots_2$

43 : 2 = 21 ; sisa 1 → d_0 *(LSB)*

21 : 2 = 10 ; sisa 1 → d_1

10 : 2 = 5 ; sisa 0 → d_2

5 : 2 = 2 ; sisa 1 → d_3

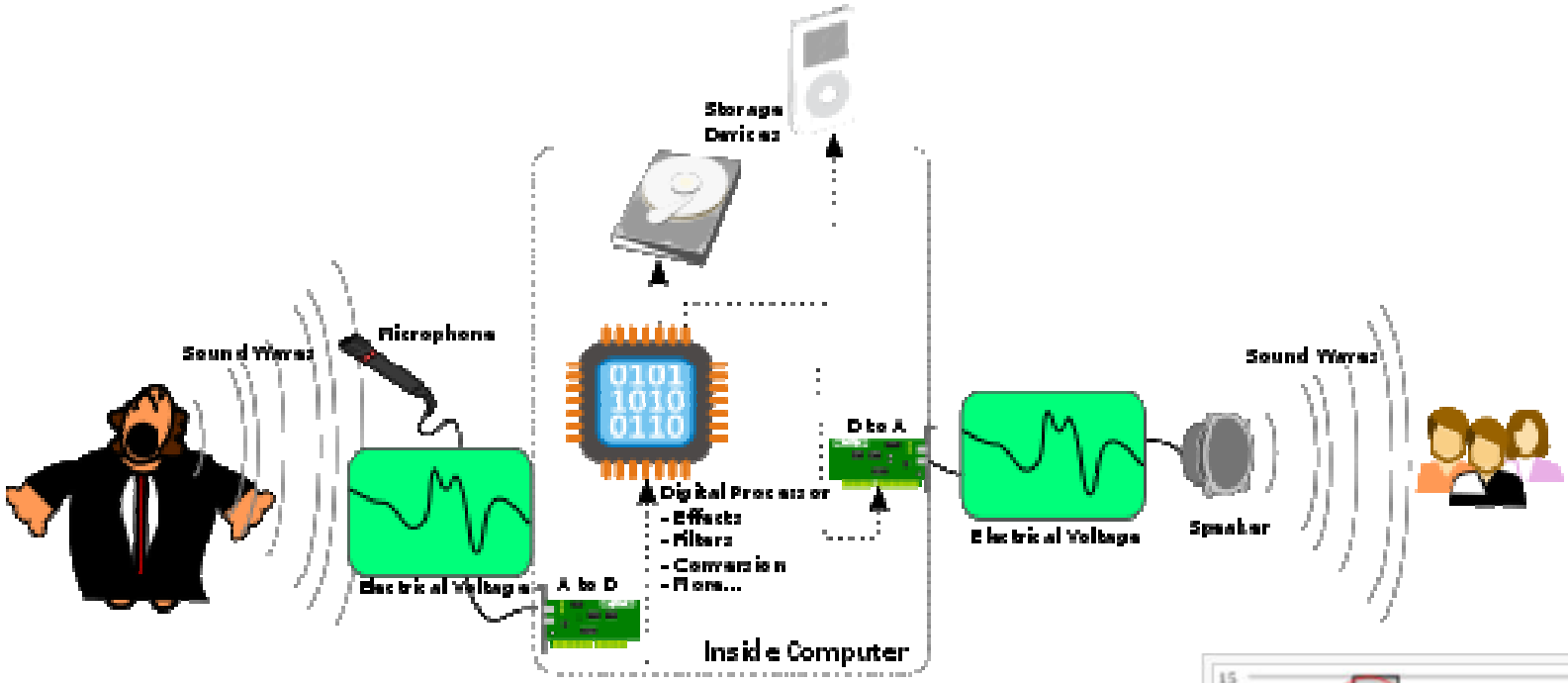
2 : 2 = 1 ; sisa 0 → d_4

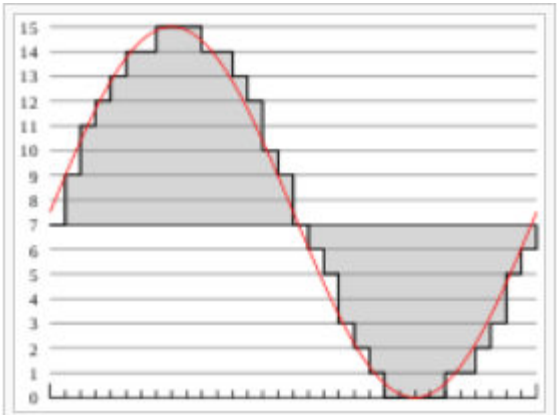
1 : 2 = 0 ; sisa 1 → d_5

Jadi $43_{10} = 101011_2$

Bagaimana cara konversi dari satu basis bilangan ke basis yang lain?
→ Basis 2, 8, 10, 16

Contoh: digital audio

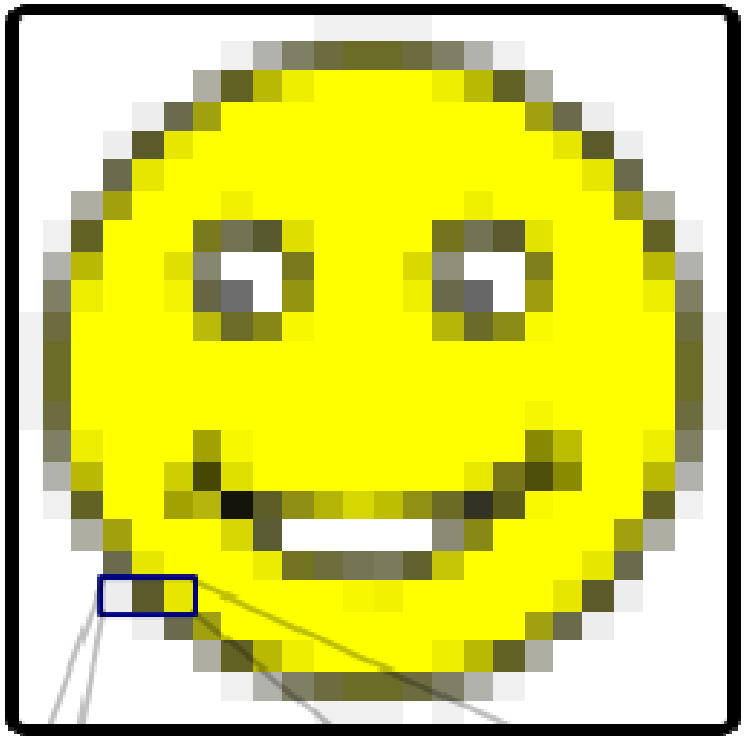


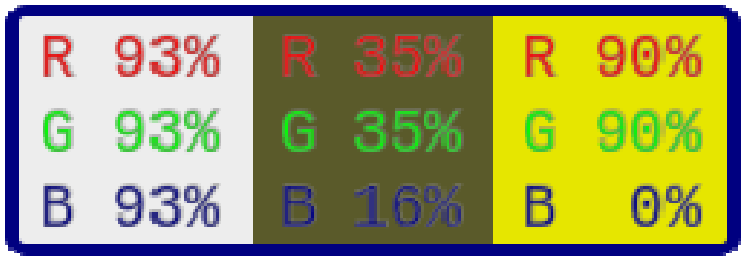


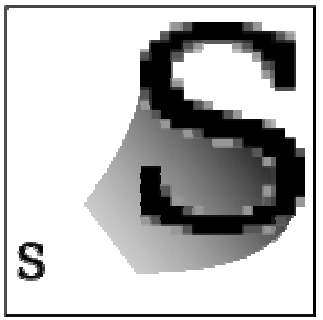
Sampling and 4-bit quantization of an analog signal (red) using Pulse-code modulation

http://en.wikipedia.org/wiki/Digital_audio

Contoh: raster image







Raster
.jpeg .gif .png

Vector
.svg

http://en.wikipedia.org/wiki/Raster_image
http://en.wikipedia.org/wiki/Scalable_Vector_Graphics

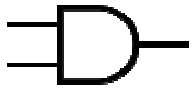
Logic ≈ Circuit

- Claude Shannon
- In a 1938 paper based upon his master's thesis at MIT, Claude Shannon demonstrated how relay and switching circuits could be expressed in the logical symbolism of the propositional calculus, and vice versa

Logic	Circuit
True	Closed
False	Open
AND	Serial
OR	parallel

Gerbang logika dasar

- AND: A.B



INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

- OR: A+B

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

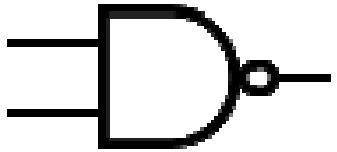
- NOT: $\overline{A}$

INPUT	OUTPUT
A	NOT A
0	1
1	0

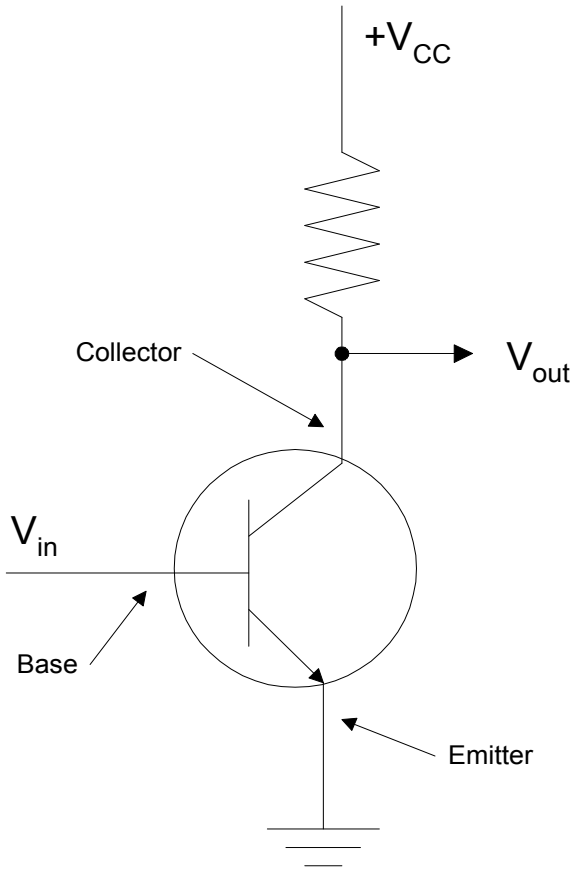
Beberapa Gerbang Logika Lain

$A \oplus B$

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Gerbang → transistor



- $V_{in} < V_{threshold}$ → transistor = resistor

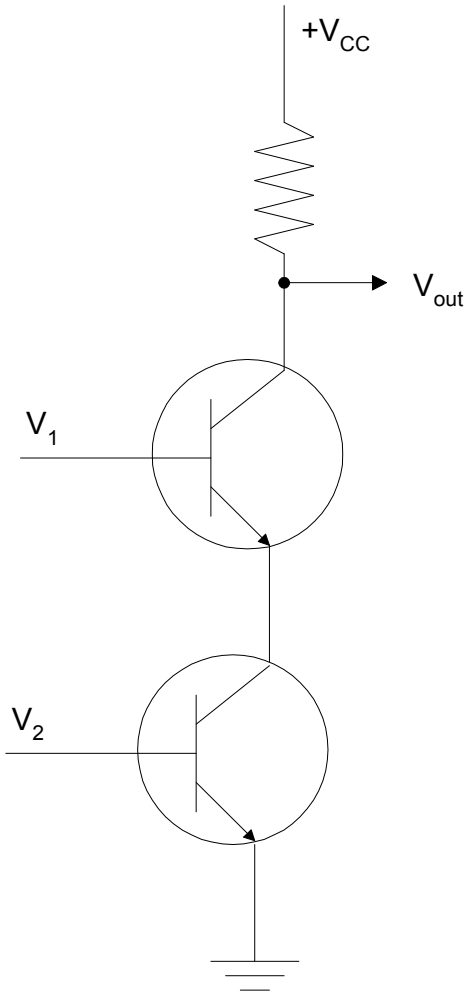
  → $V_{out} \approx V_{cc}$
- $V_{in} \geq V_{threshold}$ → transistor = konduktor

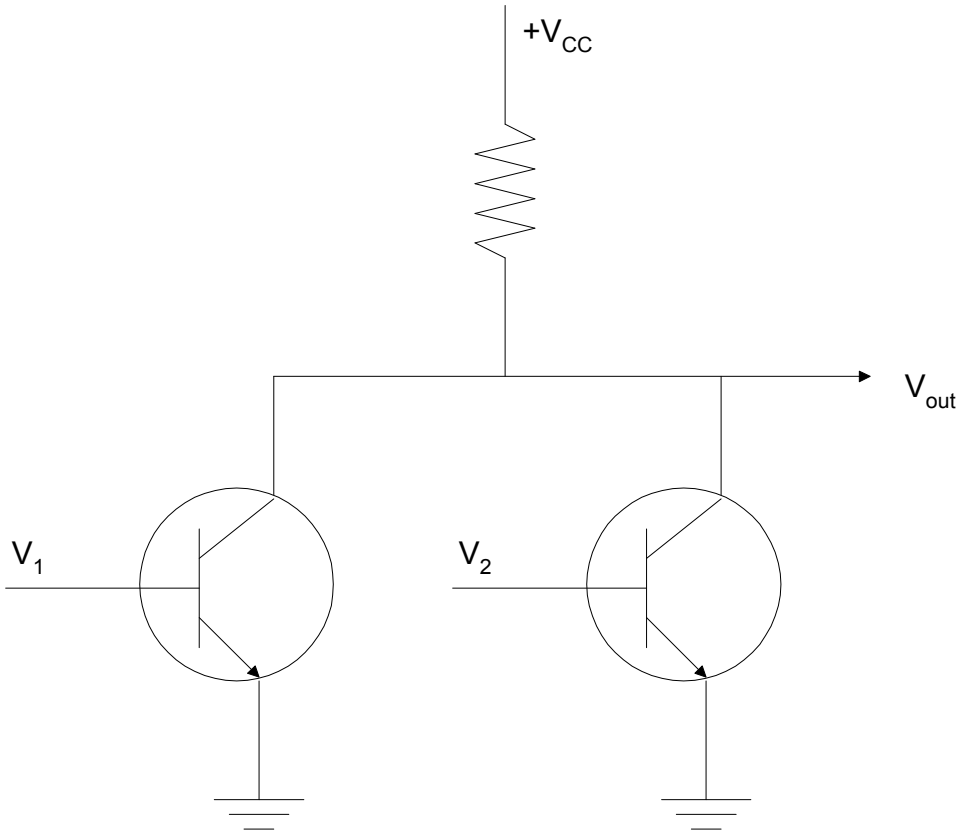
  → V_{out} = *ground*

Gerbang logika?

=> NOT (inverter)

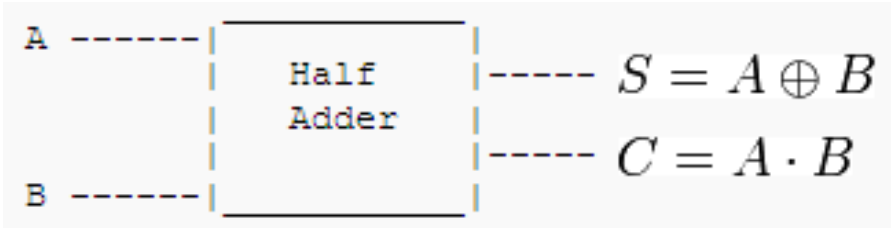
Setara dengan gerbang apa ini?

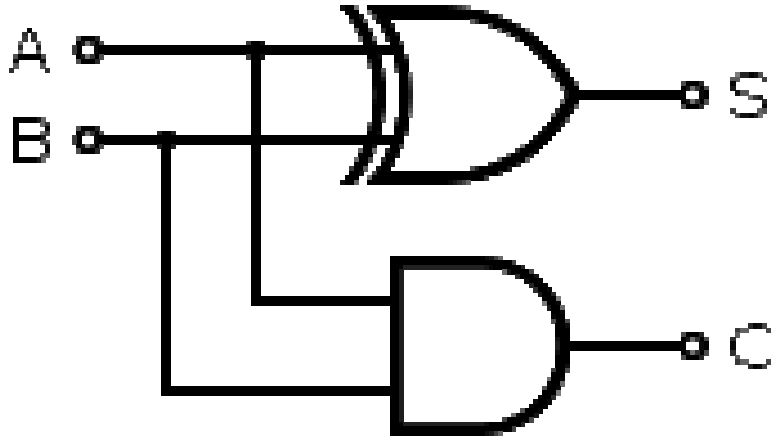




Half Adder

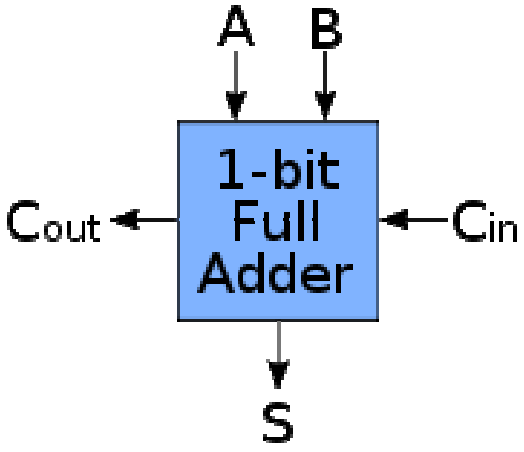
Inputs		Outputs	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0





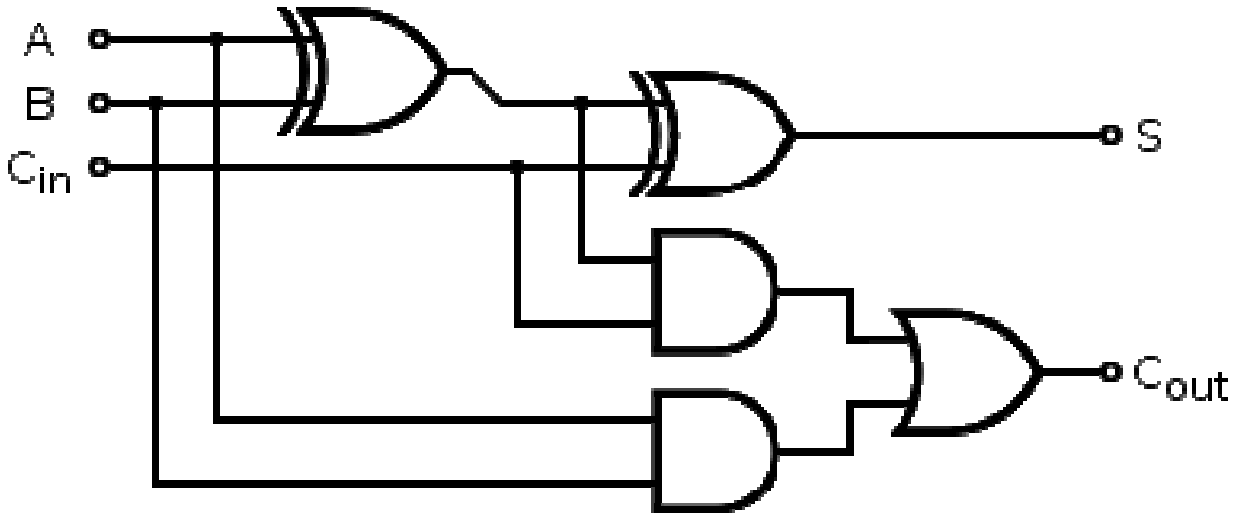
Full Adder

Inputs			Outputs	
A	B	C_i	C_o	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1



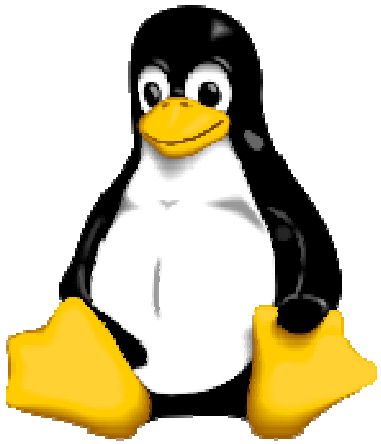
$$S = (A \oplus B) \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$$



Sistem Operasi

Thank God there are Operating Systems!



Windows

solaris

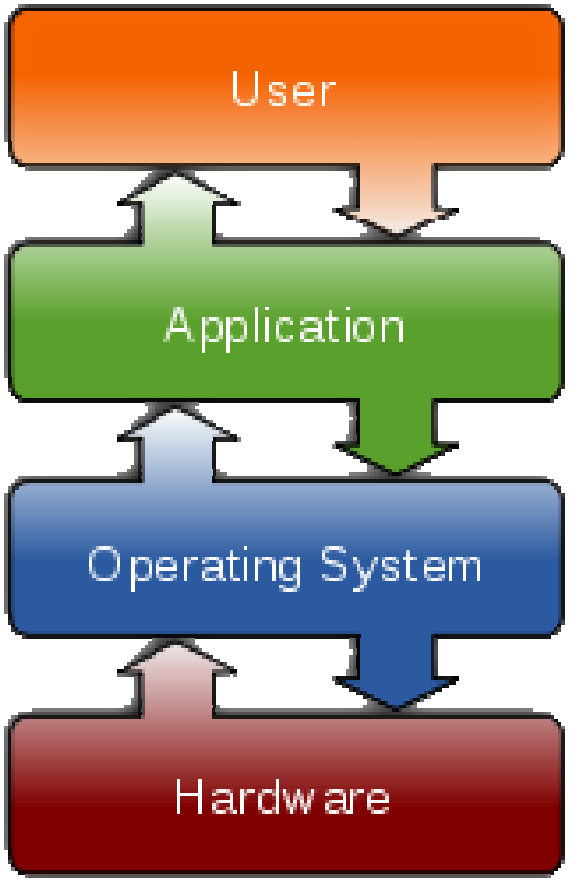
FreeBSD

AIX

etc.

Operating System is…

- an **interface** between hardware and user
- responsible for the **management** and **coordination** of activities and the **sharing** of the limited resources of the computer



OS as an…

- **Extended Machine**
  - Program that hides the truth about the hardware
  - Present nice and simple ‘machine’ to program
  - Encapsulate the detail of different physical machines

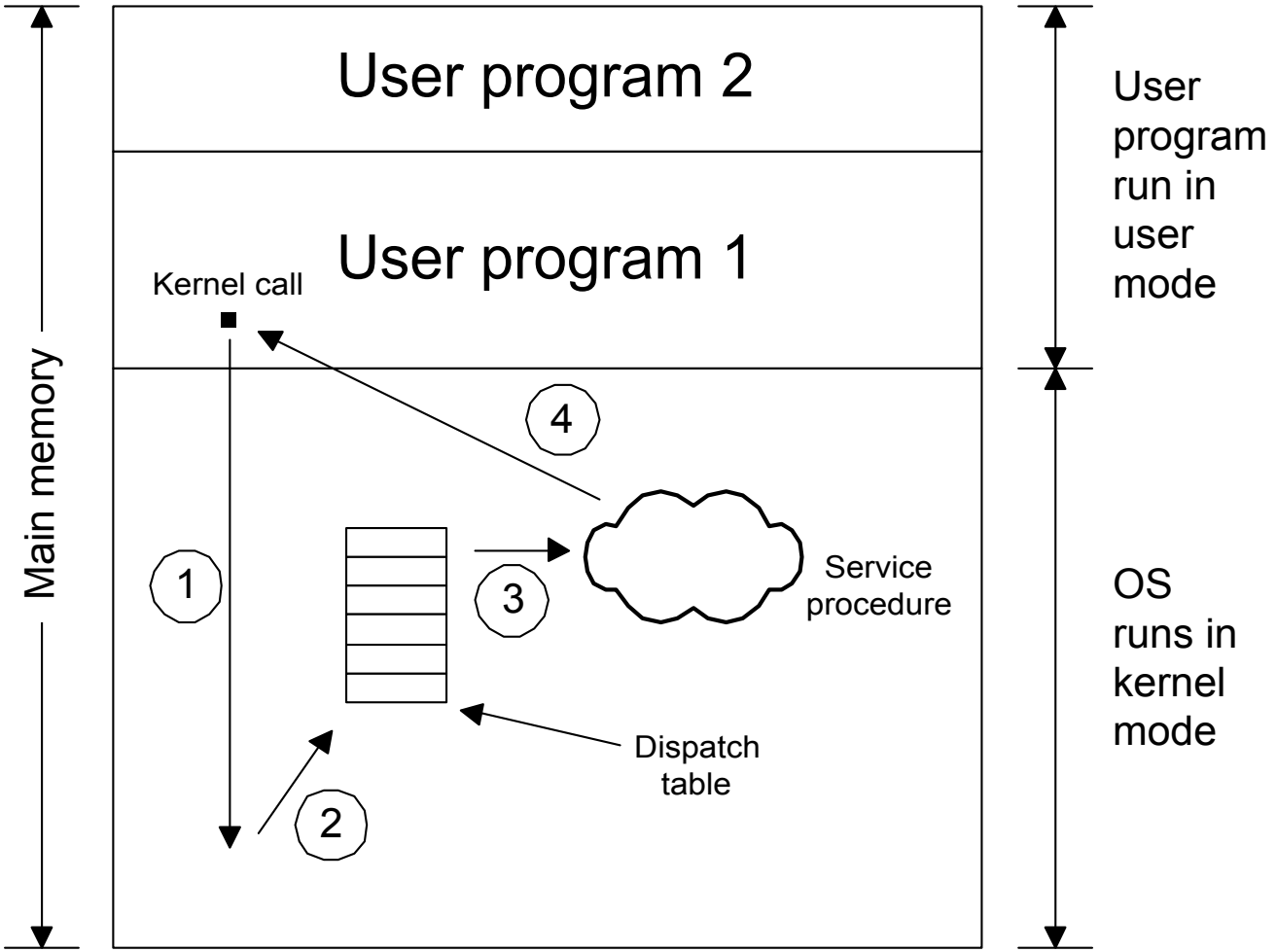
- **Resource Manager**
  - Manage all the pieces of a complex system
  - Provide orderly and controlled allocation of resources among various programs (keep track of usage, grant request, mediate conflicting request etc.)

OS Services

- OSs offer a number of **services** to application programs and users.

- Applications access these services through application programming interfaces (APIs) or system calls.
- The application can request a service from the operating system, pass parameters, and receive the results of the operation.

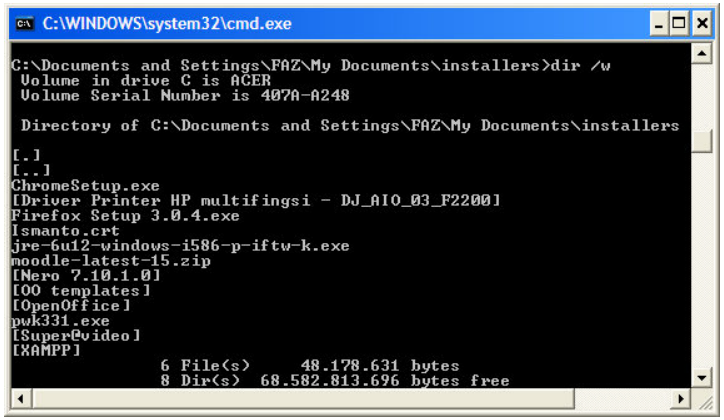
System call

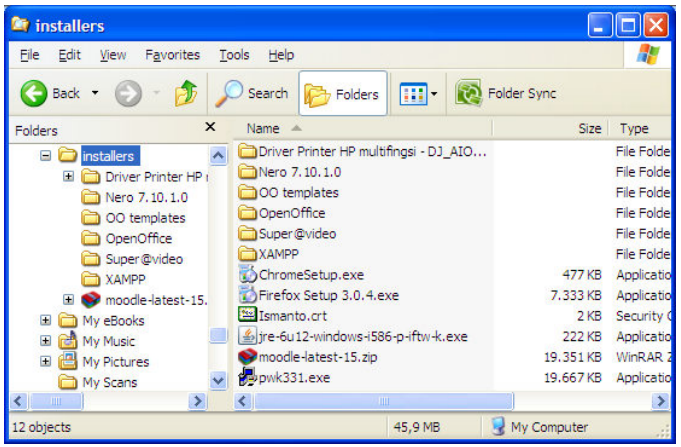


1. User program traps to the Kernel
2. Operating System determines service number required
3. Operating System locates and calls service procedure.
4. Control is returned to user program

OS-User interaction

- Users may also interact with the operating system with some kind of software user interface (UI)
  - typing commands by using command line interface (CLI)
  - using a graphical user interface (GUI)





OS History

- The first computers did not have operating systems
- early 1960s → batch processing
- Early OS on mainframes & microcomputer → only supported one program at a time
- Multitasking OS
- …

Kernel

- Core part (mandatory and common to all other software) of the operating system comprising the minimal set of functionalities
  - process management
  - memory management
  - device management
  - inter-process communication
  - system calls
  - protection
- Is always kept in main memory