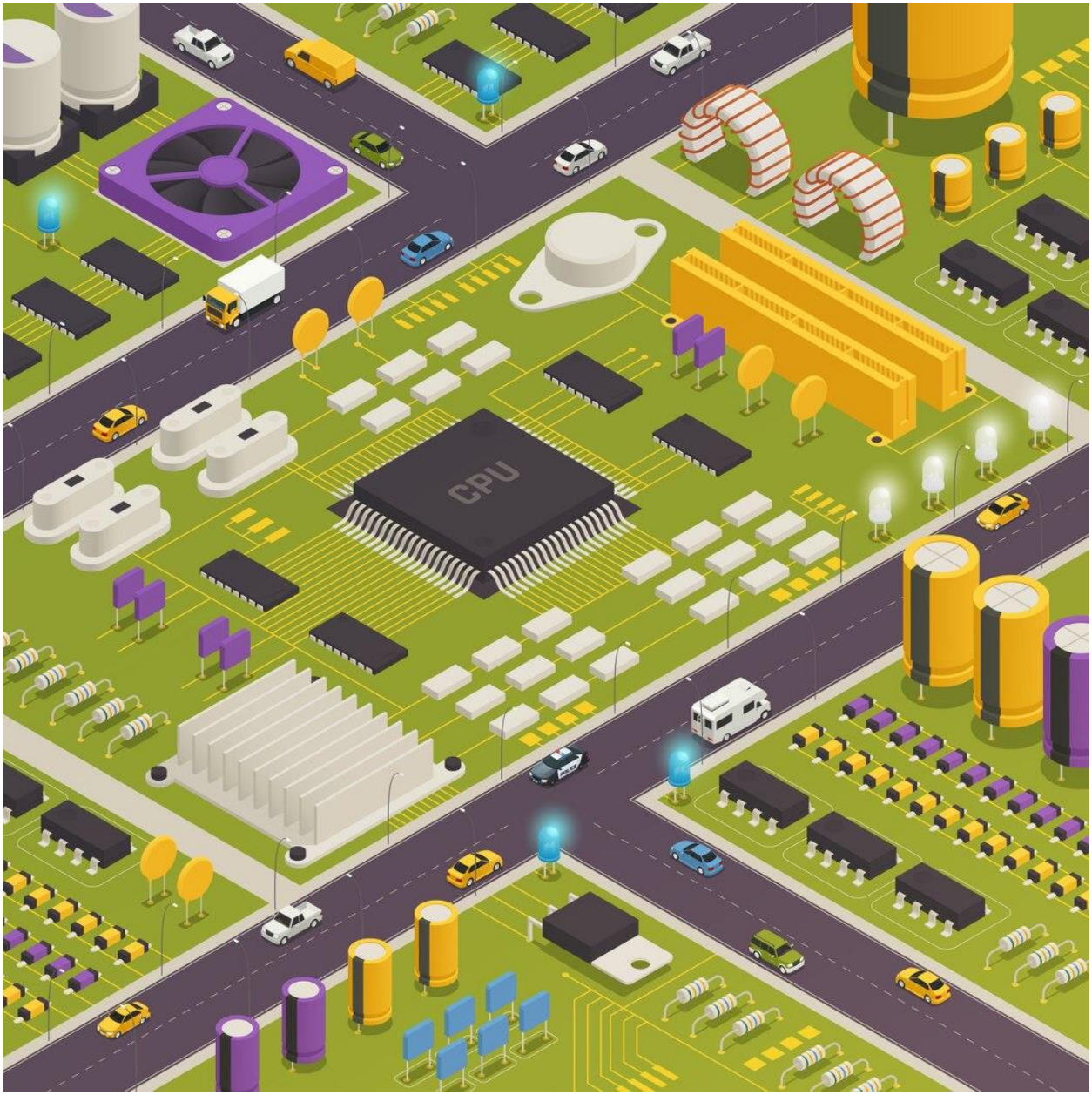
Călin BÎRĂ



==========

[Digital] Electronics by Example

When Hardware Greets Software

Preface

This book contains five chapters with material discussed in a simple manner. It will prove a good book for freshmen. By no means is this book exhaustive – it is only an attempt to put some order in the chaos created in the mind of regular high-school students. It touches subjects such as applied physics, packaging, programming, analog design, digital design, electronic devices, computer architecture but also delves into applied embedded system architecture, design, and implementation.

Chapter one provides an introduction into analog and digital signals and systems including computers and computer networks. Chapter two and three are an introduction into basic digital and analog components, while chapter four is a discussion regarding complexity. Finally, chapter five is a walkthrough on design and implementation of a few mixed-signal systems used in real-life. This book uses a lot of pictures, and instead of citing each of them with their source, I opted to create a table of figures, where I give credit where credit is due.

I thank my family, Paula, close friends, and I am deeply in debt to prof. G.M. Stefan, without whom, I would not have decided to pursue a career in academia. Special thanks go to my work colleagues and friends and to that BMW driver (you know who), to my Toyota Club friend, Mircea Laslău (what an epic entrance on 12th July morning!) and MD Anca Cîrstea without whom, I would not have had the time or capability to write or read, this book, during the summer of 2023...

Table of Contents

List of Figures

List of Tables

List of Acronyms

AACS	Advanced Access Content System
AC	Alternative Current
ACK	Acknowledge
ADC	Analog to digital converter
AES	Advanced Encryption Standard
AGM	Absorbent Glass Mat
AMD	Advanced Micro Devices
ANSI	American National Standards Institute
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
ASIC	Application Specific Integrated Circuit
BIOS	Basic Input-Output System
BJT	Bipolar Junction Transistor
BOD	Brown-out detect
CBC	Cipher Block Chaining
CD	Compact Disk
CDIP	Ceramic Dual Inline Package
CERN	Conseil Européen pour la Recherche Nucléaire / European Council for Nuclear Research
CFB	Cipher Feedback Mode
CMOS	Complementary Metal-Oxide Semiconductor
COBOL	Common Business-Oriented Language
CPHA	Clock Phase
CPOL	Clock Polarity
CPU	Central Processing Unit
CRT	Cathode-Ray Tube
CS	Chip Select
CSNET	Computer Science Network
CTFT	Continuous Time Fourier Transform
CTR	Counter
CUDA	Compute Unified Device Architecture
CV	Computer Vision
DAC	Digital to Analog Converter
DC	Direct Current
DDR	Double Data Rate
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DIP	Dual In-line Package
DNA	Deoxyribonucleic Acid
DNS	Domain Name Service

DRAM	Dynamic Random Access Memory
DUT	Device Under Test
EB	Exabyte
ECB	Electronic Code Book
ECC	Error Correction Code
EMES	Engineering of Modern Electric Systems
FET	Field-Effect Transistor
FIFO	First-In First-Out
FIPS	Federal Information Processing Standard
FM	Frequency Modulation
FTP	File Transfer Protocol
GB	Gigabyte
GHz	Gigahertz
GND	Ground
GP	General Purpose
GPIO	General Purpose Input/Output
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HD	High Definition
HDD	Hard Disk Drive
HDL	Hardware Description Language
HTML	Hypertext Markup Language
I2C	Inter-integrated circuit
IC	Integrated circuit
ICPSC	International Conference on Signal Processing and Communication
IDE	Integrated development environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IIC	Inter-integrated circuit
IOP	I/O operation
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
IR	Infrared
ISO	International Organization for Standardization
JBOD	Just a Bunch of Disks
JEDEC	Joint Electron Device Engineering Council
JS	JavaScript
KB	Kilobyte
kHz	Kilohertz
LAN	Local Area Network
LASER	Light Amplification by Stimulated Emission of Radiation

LCCC	Leaded Chip Carrier ceramic
LED	Light-emitting diode
LIDAR	Light Detection and Ranging
LSB	Least Significant Bit/Byte
MATLAB	Matrix Laboratory
MB	Megabyte
MCU	Microcontroller Unit
MHz	Megahertz
MIL-STD	Military Standard
MISO	Master In Slave Out
MIT	Massachusetts Institute of Technology
MOhm	Megohm
MOS	Metal-Oxide Semiconductor
MOSI	Master Out Slave In
MSB	Most Significand Bit/Byte
MW	Megawatt
NACK	Not Acknowledge
NAND	Not AND
NAS	Network-Attached Storage
NC	Normal Closed
NI	National Instruments
NIC	Network Interface Card
NIH	National Institute of Health
NIMH	Nickel–metal hydride (battery)
NIST	National Institute of Standards and Technology
NO	Normal Open
NOM	Nominal
NOR	Not OR
NP	Non-polynomial
NSF	National Science Foundation
NTC	Negative Temperature Coefficient
NTP	Network Time Protocol
NVM	Non-Volatile Memory
OFB	Output Feedback Mode
OOK	On/Off Keying
OP	Operation
OSI	Open Systems Interconnection Model
OTP	One Time Password
OUT	Output
PB	Petabyte
PC	Personal Computer

PCB	Printed Circuit Board
PCBC	Propagating or Plaintext Cipher-Block Chaining
PCI	Peripheral Component Interconnect
PDIP	Plastic Dual Inline Package
PHP	PHP: Hypertext Preprocessor
PIR	Passive Infrared
PN	Part Number
PTC	Positive Temperature Coefficient
PTP	Picture Transfer Protocol
QSPI	Quad SPI
R,G,B	Red, Green, Blue
RADAR	Radio Detection and Ranging
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RC	Remote Control
RDS(on)	Resistance from drain to source, when in on state
RF	Radio Frequency
RFC	Request for Comments
RFID	Radio Frequency Identification
RGB	Red Green Blue
RH	Relative humidity
RMS	Root Means Square
RNA	Ribonucleic Acid
RSA	Rivest-Shamir-Adleman (Encryption)
RX	Receiver or reception
SAR	Successive approximative register
SATA	Serial Advanced Technology Attachment
SCK	Serial Clock
SCL	Serial Clock
SCR	Silicon controlled rectifier
SD	Secure Digital
SDA	Serial Data
SDRAM	Synchronous Dynamic Random Access Memory
SFTP	Secure File Transfer Protocol
SMD	Surface Mount Device
SNTP	Simple Network Time Protocol
SOIC	Small Outline Integrated Circuit
SPDT	Single Pole Double Throw
SPI	Serial Peripheral Interface
SPST	Single Pole Single Throw
SQL	Structured Query Language

SRAM	Static Random Access Memory
SS	Slave Select
SSD	Solid State Drive
SSH	Secure Shell
STFT	Short Term Fourier Transform
TB	Terabyte
TCP	Transmission Control Protocol
TCR	Temperature Coefficient of Resistance
TDP	Thermal Design Power
THT	Through Hole Technology
TIOBE	The Importance Of Being Earnest
TTL	Transistor-to-transistor logic
TV	Television
TVS	Transient Voltage Suppressors
TWI	Two Wire Interface
TX	Transmitter / Transmission
UART	Universal Asynchronous Receiver/Transmitter
UDIMM	Unbuffered Dual In-Line Memory Module
UDP	User Datagram Protocol
UHF	Ultra-High Frequencies
URL	Uniform Resource Locator
US	United States
USA	United States of America
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
UV	Ultraviolet
V	Volt
VHDL	VHSIC Hardware Description Language
VHF	Very High Frequency
VI	Input Voltage
VIH	Input Voltage (high, minimum)
VIL	Input Voltage (low, maximum)
VO	Output Voltage
VOH	Output Voltage (high, minimum)
VOL	Output Voltage (low, maximum)
WAN	Wide-Area Network
WDT	Watchdog Timer
WWW	World-Wide Web
XOR	Exclusive OR
YB	Yottabyte
ZB	Zettabyte

1. Brief Introduction to Digital Circuits and Programming

1.1 Analog signals

All the studied systems in high-school's physics classes were composed from analog equipment/devices (voltage supplies, current supplies, resistors, capacitors, inductors, lightbulbs etc.). They are circuits where, for example, the voltage varies continuously within some limits. A default system is an analog audio amplifier, which takes an analogue audio signal, amplifies it (keeps the shape, but delivers more power from the supply) and sends it to the speakers. Analogue signals are hard to store and process, so lately, digital signals are used increasingly.

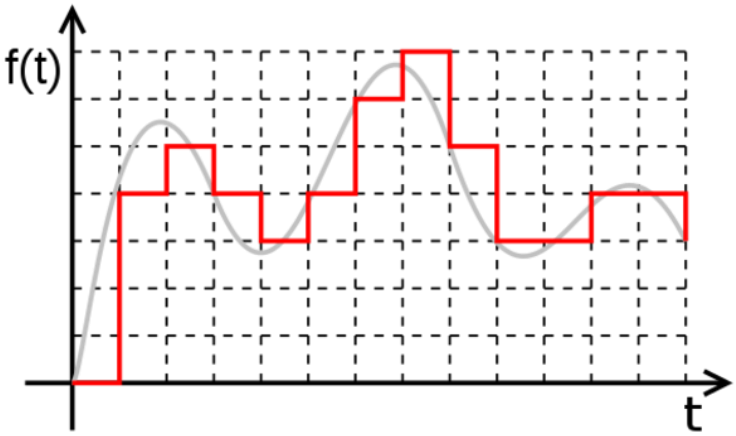


Figure 1. A digital signal (as a result of both sampling and quantization processes)

1.2 Digital signals

A digital signal is a signal which is discrete (as opposed to continuous) in both time and value. To create a time-discrete signal, one samples a continuous one. To create a value-discrete signal, one quantizes a continuous one.

The number of samples taken in a unit of time is called sampling rate (e.g., 44100 Hz == samples per second, CD-quality). The number of bits (0 or 1 symbols) required to express the amplitude is linked directly to the number of quantization steps (e.g., 16-bit for 2 to the power of 16 = 65536 steps, in the case of CD-audio quality)

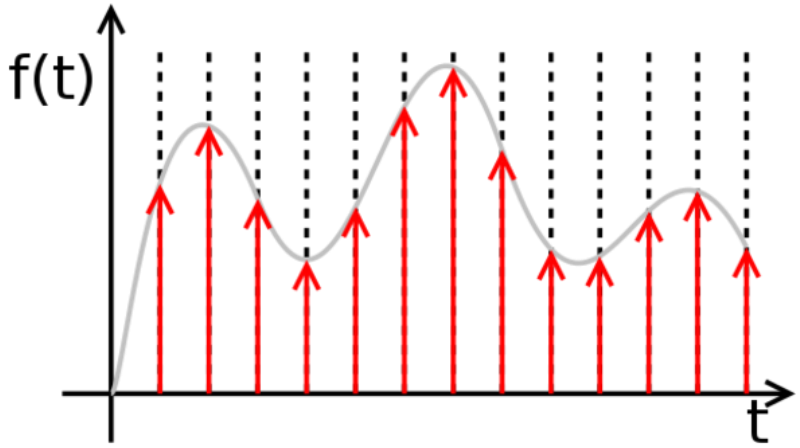


Figure 2. A time-sampled signal (as a result of sampling process)

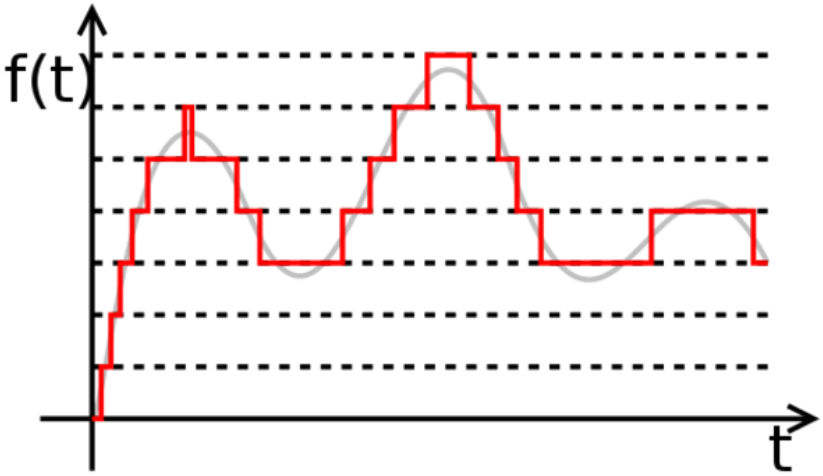


Figure 3. A value-sampled signal (as a result of quantization process)

Digital signals are used because their expression uses numbers, which allows easy storage, copying (without loss) and processing. Immunity to noise can be obtained using mathematical instruments like error-detection and error-recovery processes. In addition, the quality (how similar it looks to the source analogue signal) can be chosen as a compromise. The more quantization steps we use (e.g. infinite) the more accurate the value is to the source signal's value: however, these systems are usually used for the comfort of human life, so the trade-off will take into consideration human hearing or human sight etc. which will not push the quantization step too high (e.g. audio signals are good enough when using 64k steps, that is 16-bits per sample; video signals are good enough when colors are represented with 256 steps of Red, Green and Blue, therefore 3x 8-bits are enough for a pixel) Regarding

the recovery of a continuous signal from the time-sampled signal, we have the sampling theorem which demonstrates that we can fully recover the original, as long as the sampling rate is at least twice the maximum frequency contained in the original signal. For example, if one wants to recover up to 22 kHz audio signals (more than what the common human ear can hear), one should use 44 kHz sampling rate.

1.3 Digit, number, and radix

A radix-10 number uses a dictionary of 10 symbols (the ten digits) to express any number. For example, number 123 is made of 1 * 100 + 2 * 10 + 3. The radix of 10 is not the only known radix but is the most used by humans (arguably because we have ten fingers and can count easily using them). However, to use radix of 10, one must distinguish between 10 different symbols (0-9). Digital electronics use radix of 2 because it is easier to distinguish between only two symbols, therefore it is easier to store information in this form. The trade-offs that same number expressed in radix of 10 is around 3.5 times shorter that a radix of 2. For example, 9 is expressed in radix 2 with the sequence 1001, the number 127 is 11111111 etc. The symbols available for the radix of 2 are 0 and 1, and they are called BInary digiTS, in short bits. Using 2 digits we can express numbers from 0 to 99 (that is, 100 different numbers). Using 2 bits we can express numbers from 0 to 3 (that is, 4 different numbers). Most common radices are 2 (binary), 8 (octal), 16 (hex), 10 (dec) and 256. We will mark numbers in radix 2, as prefixed with 0b e.g.: 0b1001 is number 9 in radix 10. The hexadecimal number will be prefixed with 0x e.g.: 0x10 is 16 in radix 10.

Table 1. Multiples of bits / bytes according to JEDEC [1]. IEC 80000-13 standard changes the name for the power of two, by inserting a "bi" in the name: Kibibyte, Mebibyte, Gibibyte.

Memory Unit (JEDEC)	**Memory unit (IEC)**	**Description**
Bit	Bit	Binary Digit 1 or 0
Kbit	Kibibit	1024 bits
Mbit	Mebibit	1024 Kbits
Byte	Byte	8 bits
KiloByte(KB)	KibiByte (KiB)	1024 Bytes
MegaByte(MB)	MebiByte (MiB)	1024 KB
GigaByte(GB)	Gibibyte (GiB)	1024 MB
TeraByte(TB)	Tebibyte (TiB)	1024 GB
PetaByte(PB)	Pebibyte (PiB)	1024 TB
HexaByte or exaByte (EB)	Exbibyte (EiB)	1024 PB
ZettaByte (ZB)	Zebibyte (ZiB)	1024 EB
YottaByte (YB)	Yobiibyte (YiB)	1024 ZB

1.4 Digital systems

Digital systems are designed to store and process and exchange information in digital form. They are found in a wide range of applications, including process control, communication systems, digital instruments, and consumer products. These systems/circuits may be classified by the number of *appropriate* loops enclosed within [3]; more loops will mean more autonomy, therefore *smarter* circuits.

0 - loop circuits: contain only combinational circuits (logic gates)

1 - loop circuits: the memory circuits, with behavioral autonomy in their own internal states; they are mainly used for *storing*

2 - loops circuits: the automata, with the behavioral autonomy in their own state space, performing mainly the function of *sequencing*

3 - loops circuits: the processors, with the autonomy in interpreting their own internal states; they perform the function of *controlling*

4 - loops circuits: the computers, which interpret autonomously the programs according to the internal *data*

n-loop circuits: systems in which the information is interpenetrated with the physical structures involved in processing it; the distinction between *data* and *programs* is surpassed and the main novelty is the *self-organizing* behavior.

Any k-loop circuit can do everything any k-1 loop circuit can do.

While 0 – loop circuits (combinational logic circuits) are quite easy to grasp as they are very simple in structure and behavior, the more evolved circuits, containing sequential circuits (with the clock signal driving them) are the ones used to handle complexity.

Some common 0-loop circuits are: logic gates, multiplexers (sends the selected digital input to the output), demultiplexers (send the input to the selected output), decoders (sends logic 1 to the selected output), adders, subtractors, ALUs (arithmetical-logical units), equality comparators, magnitude comparators etc.

Some common sequential circuits are flip-flops (FFs), registers, counters/timers, and FSMs.

For an in-depth discussion on this subject, refer to book [3].

1.5 Programming a 4-loops digital system (computer)

Writing a program for a computer requires describing steps to be performed in a specific order. The description is done in a "programming language" usually containing keywords in English language (like C, Java, Python), written in a source file, then interpreted or compiled and ran on a computer.

1.6 Components of a computer

Motherboard: the mainboard that electrically links all high-speed components of the PC

- CPU: central processing unit, the "brain" of the computer
- GPU: graphical processing unit, which renders images to be shown on the monitor.
- RAM: random-access memory, is used while running programs
- Storage: long-term storage (non-volatile storage) where operating systems / applications / files reside
- Usual peripherals:
    - Monitor (display) - where people may see visually the output of the computer.
    - Mouse (hand-held pointing device, that detects 2D relative movement on a surface)
    - Keyboard (key-based input device, usually holds all 26 English letters, 0-9 digits and punctuation marks)

Figure 4. AMD Athlon64 x2 CPU (left is top, right is bottom) with Socket 939

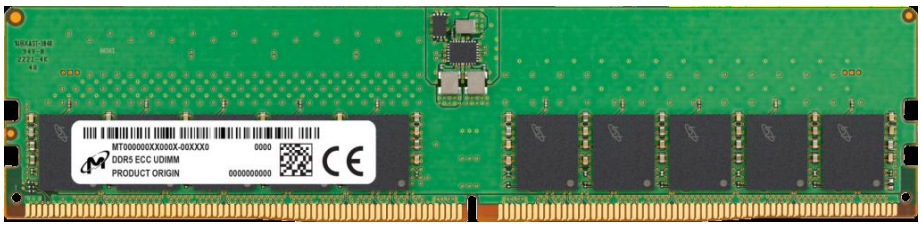


Figure 5. 32GB DDR5-4800 RAM server memory with error correction code (ECC)

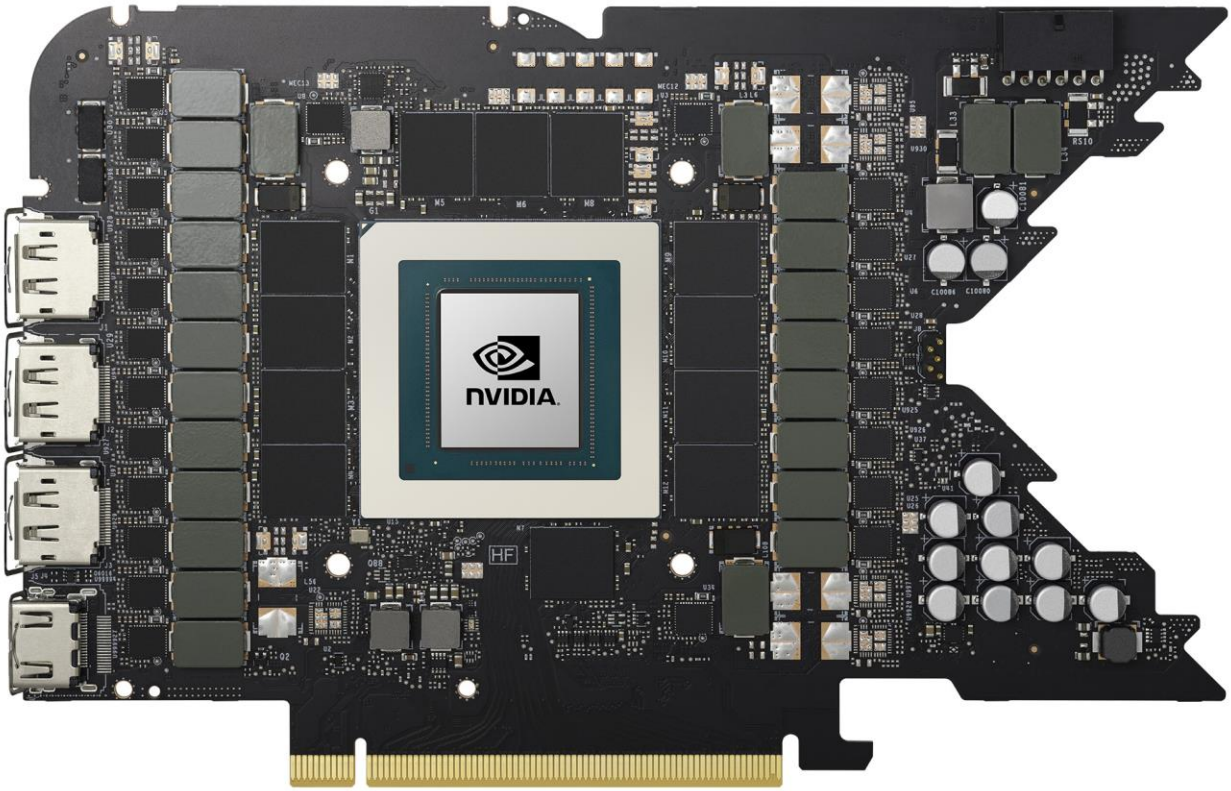


Figure 6. RTX4090 GPU card with PCIexpress x16 slot, without passive and active cooling installed.

The GPU chip is the largest square chip, the black vertical parts (10+) are coils for the power lines and the 12 black chips around the GPU are memory chips

Figure 7. A SATA-interfaced SSD (Solid State Disk) for non-volatile storage.

Data rate (read and write speed) is limited by the SATA3 interface at up to 550 MB/s, making them slower than the PCIexpress versions. As a transfer rate, they can go up to 7 GB/s for both read and write)

Figure 8. A PCIe-interfaced SSD for long term storage.

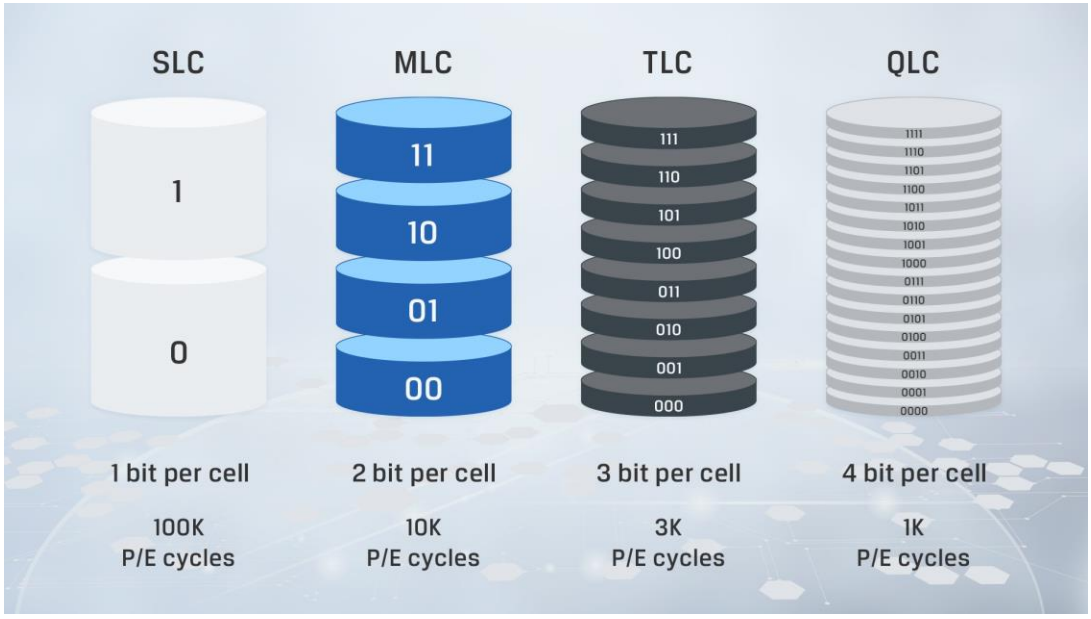


Figure 9. A comparison between SLC, MLC, TLC and QLC cells

Random access non-volatile storage is provided by electro-mechanical or electronic storage. While the mechanical hard drives had their advantages in capacity, nowadays, electronic storage caught up and enable movement-free information storage. There are multiple types of cells used in solid-state drives, each having their compromise between cost, capacity and endurance. High endurance cells (SLC) allow only one bit of data to be stored in a cell (which provides large noise margin), whereas all other cells decrease the noise margin to increase data density. A common SLC memory allows over 100K cycles of programming (set to 0) or erase (set to 1), whereas QLC only handles thousands.

Monitors advanced from the CRT (cathode-ray tube) technology where a electrons emitted from a source are directed using electromagnets towards the surface of the screen, to flat-tube technologies where light is either backlight is let pass through a controllable array of pixels, or the very pixels (picture elements) are composed of individual LEDs which are powered on and off selectively, by the same commands (as CRTs) coming from a video card.

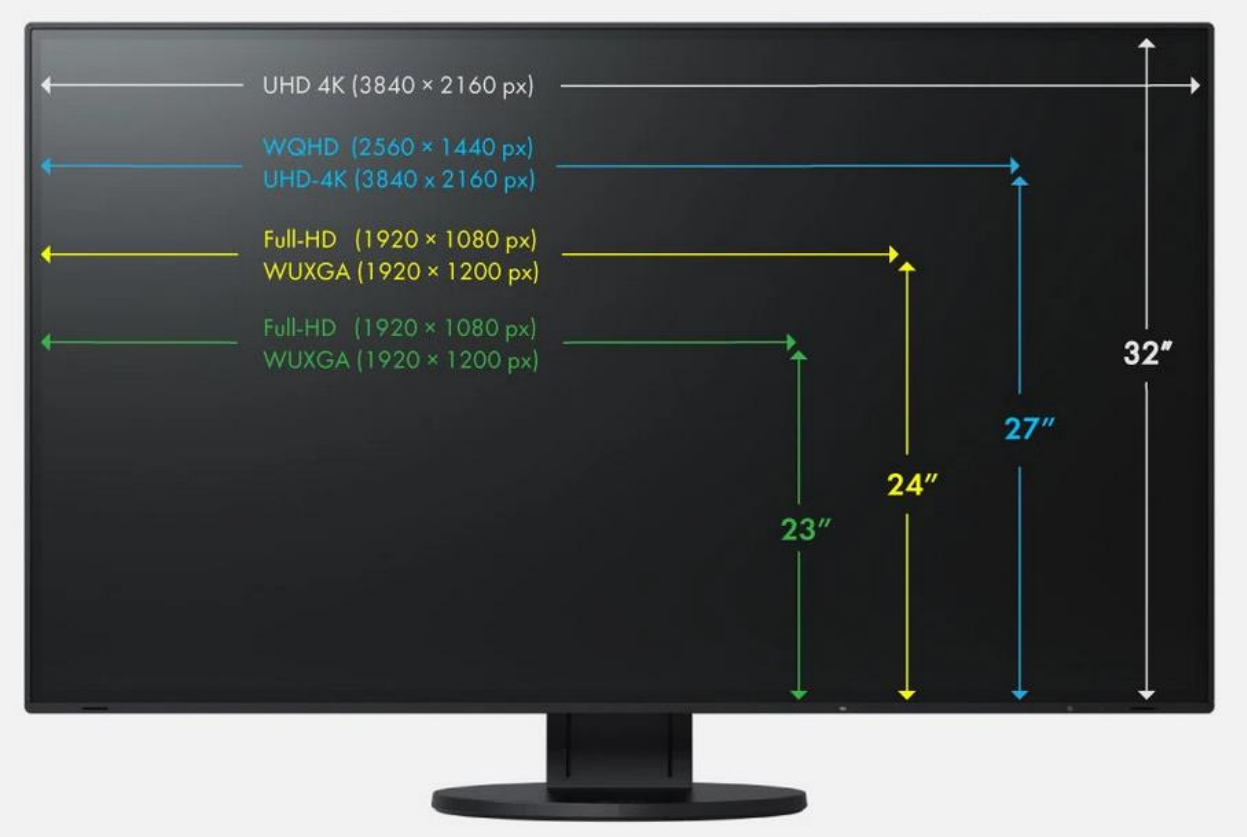


Figure 10. A monitor with common pixel resolutions (HD = High definition).

The monitor surface is composed of pixels (individual light sources/sinks, that are lit independently according to the image sent by the software using the video card, via video interface). For 4K monitors, there are about 8 million pixels in each image. To create the effect of a moving picture, the images are time-spaced at 33 milliseconds (30 frames per second), 16.6ms (60 fps) or less, for gaming setups (where higher fps allows faster reactions)

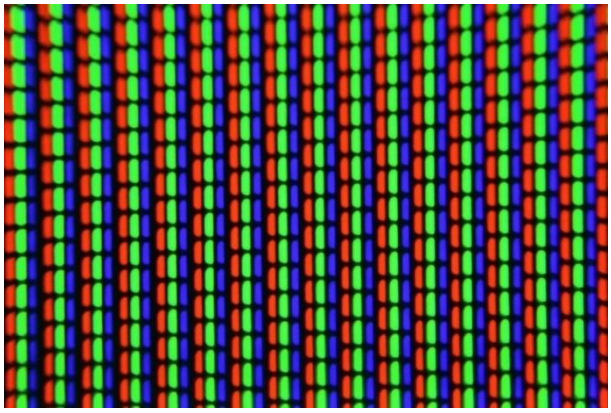


Figure 11. RGB (red, green, blue) pixels in monitor image that spatially combine to obtain all colors (white color is all RGB leds in that pixel lit)

RAID

The RAID (Redundant array of independent/inexpensive disks) is a way of grouping drives into more flexible setups for larger storage, higher speed or higher redundancy. In the past, when SSD were not present, mechanical drives were the only way for large non-volatile, random-access storage. This implies that mechanical movement of the read and write head, limited the read/write speed of the drive. The server drives were smaller in diameter (to allow a smaller movement) and higher in speed (regular drives were 5400 or 7200 rpms, whereas server drives were 10k or 15k rpms). The emergence of SSDs with their electronic storage cancels the speed advantages of the RAID arrays using mechanical drives, however enthusiasts may still group SSDs into RAID arrays for data redundancy purposes.

Table 2. Common RAID modes and their minimum requirements

RAID mode	Minimum number of disks
JBOD	1
RAID 0	2
RAID 1	1 (no data protection) 2 (with data protection)
RAID 5	3
RAID 6	4
RAID 10	4

1.6.1.1 JBOD (Just a bunch of disks)

JBOD is a configuration where data is stored sequentially: data not fitting the first disk is automatically overflown into second disk etc.

Advantage: sum of the disks space is the total available disk space.
Disadvantage: any one drive fails implies data will be lost
Total capacity of N drives with C storage capacity each: N * C

1.6.1.2 RAID 0 (data striping)

The RAID 0 mode is the fastest RAID mode as it writes data consecutive data blocks on different disks, which, in mechanical drives means the access time is decreased, therefore increasing the read and write data rate. It is the preferred choice for gamers.

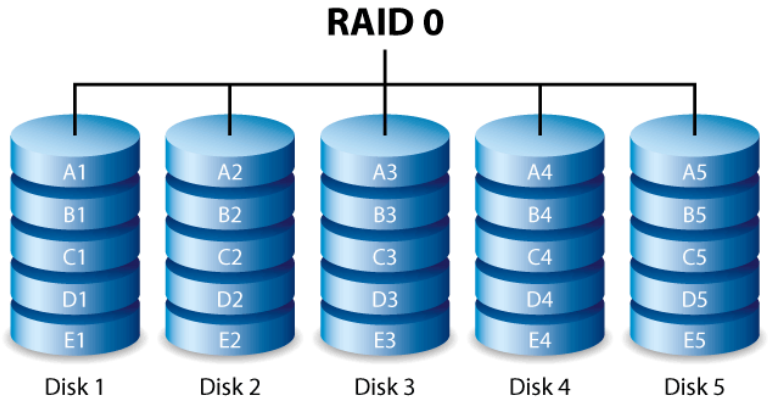


Figure 12. RAID0: high speed, no redundancy

Advantage: high read/write/data speed, higher that single-drive
Disadvantage: if one drive fails, the array fails
Total capacity of N drives with C storage capacity each: N * C

1.6.1.3 RAID 1 (data mirroring)

The RAID 1 mode mirrors all data on all drives. Assuming the array is made of N drives of C storage capacity each, the available capacity of the array is C but the reliability is very high, with some speed improvements when reading a lot of small files.

Advantage: for N-drive array, all N drives must break to break the array
Disadvantage: expensive (high cost per available storage capacity)
Total capacity of N drives with C storage capacity each: C

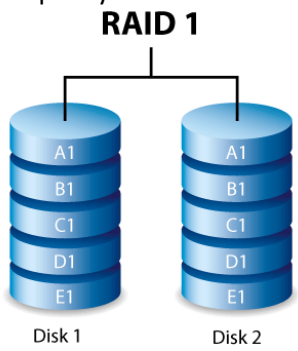


Figure 13. RAID1 mode: data mirroring

1.6.1.4 RAID5

The RAID5 is a compromise between the speed and reliability. A parity block is written consecutively on each drive, along a series of data blocks. The parity block allows data reconstruction if any one drive fails.

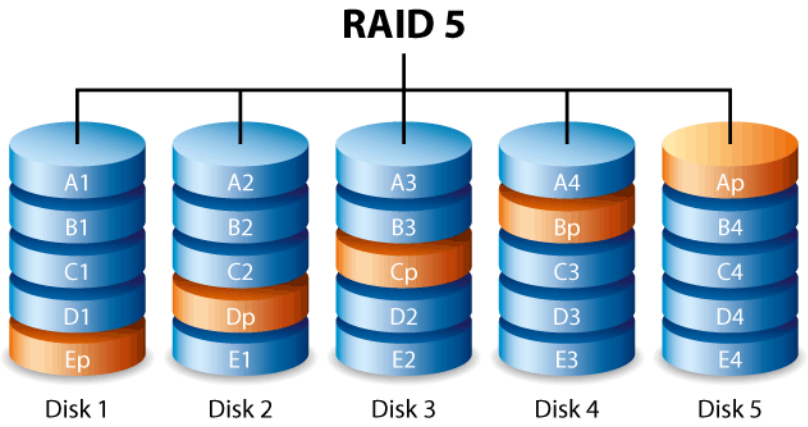


Figure 14. RAID5 mode: single drive redundancy

Advantage: one drive may fail and the array will still not lose data
Disadvantage: a drive's capacity is not available in the array, since it is reserved for parity
Total capacity of N drives with C storage capacity each: (N-1) * C

1.6.1.4 RAID6

As opposed to RAID5, where only one parity block is written per each set of data blocks, in RAID6, two parity blocks are written: this increased redundancy but decreases the available storage space.

Advantage: two drives are allowed to fail and the array will not lose data
Disadvantage: two drives' capacity is not available since it is reserved for parity
Total capacity of N drives with C storage capacity each: (N-2) * C

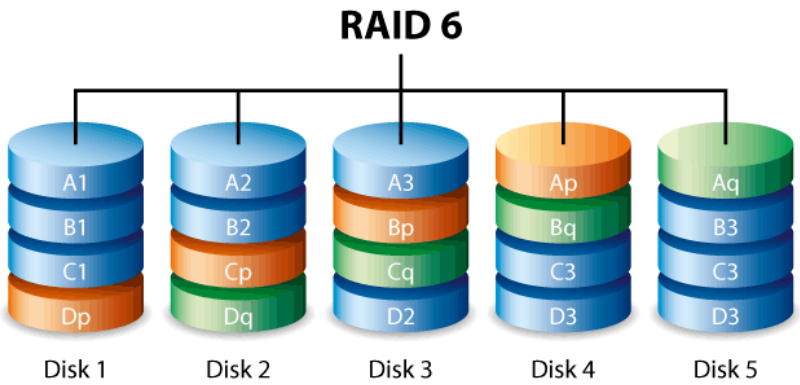


Figure 15. RAID6 mode: dual drive redundancy

1.6.1.5 RAID10

RAID10 is a combination of RAID1 and RAID0 to increase performance yet having high data redundancy. It is used mainly for increasing performance in accessing small files and keeping high reliability of the array.

Advantage: speed, and up to half of the drives are allowed to fail and the array might not lose data
Disadvantage: half of storage space is unusable
Total capacity of N drives with C storage capacity each: (N/2) * C

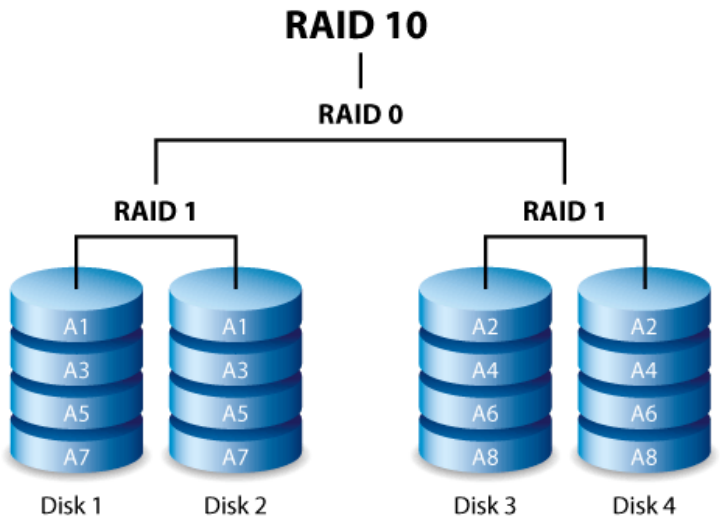


Figure 16. RAID10: a RAID 0 of RAID1s

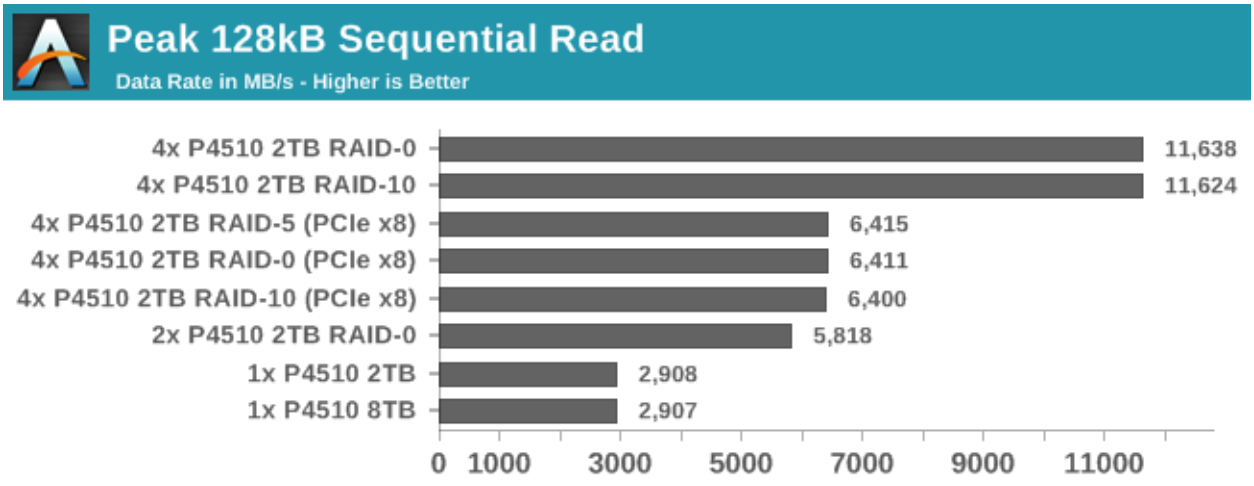


Figure 17. Anandtech's comparison of Intel P4510 SSD (x4 PCIe) drives in various RAID configurations

1.7 Computer networks

Computer networks are based on a layered model, where each level does its job requested by the higher layer and forwards new work to the lower layer. This ensures that no layer is too complicated, and multiple technologies of the same type may be mixed and matched easily.

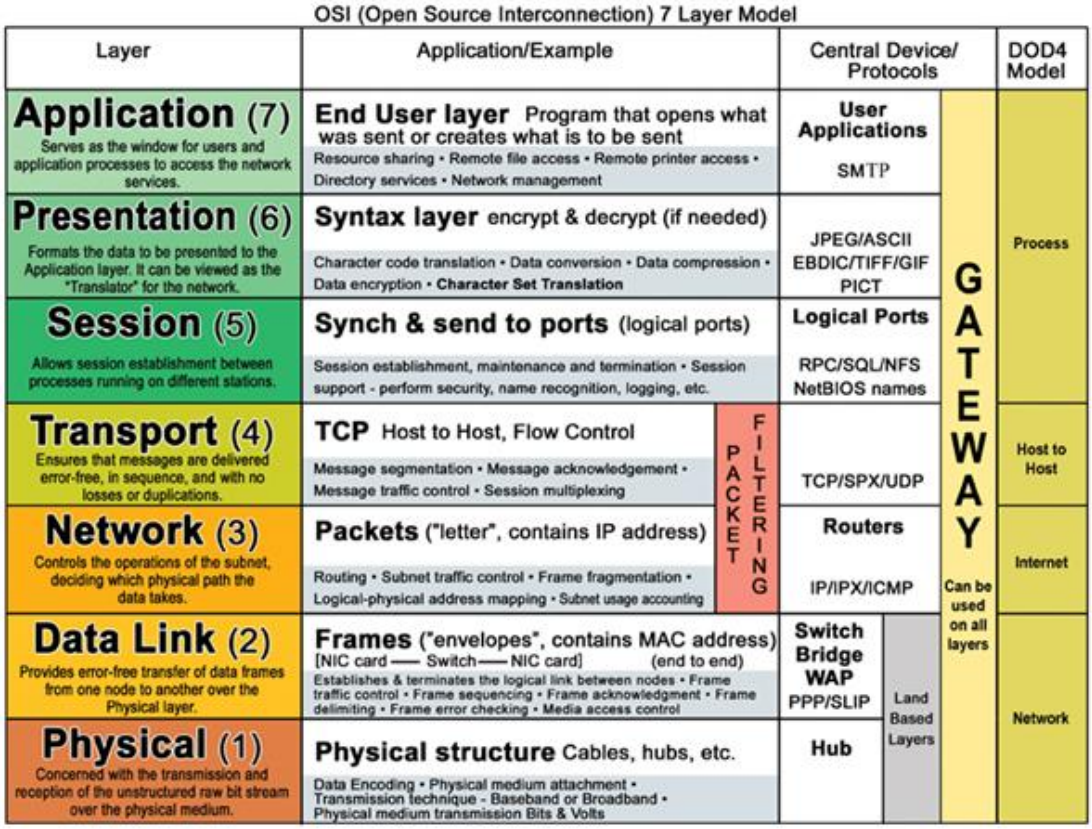


Figure 18. OSI layer model for networks

Hardware components of a computer network are the ones implementing Layer 1 (cables, connectors), Layer 2 (switches), Layer 3 (routers). From Layer 4 upwards, the software running on a computer, will take over.

Software components of a computer network are the ones implementing Layer 4 (data protocols), Layer 5 (ports/sockets), Layer 6 and Layer 7 (applications)

The tiered approach allows a network to change for example the physical layer, by replacing the NIC (network interface card) from one implementing copper wires to fiberoptics for example, without changing the upper layers (including the application).

Common transport protocols for sending data are: TCP (Transport Control Protocol, see RFC 9293) and UDP (Unified Datagram Protocol, see RFC 768). TCP has the advantage of guaranteeing delivery and the order of the packets (by error detection and correction via

checksum, ack and timeout) and is used for example if file transfers, whereas UDP has the advantage of low-latency (and is mainly used when obsolete data that was not transmitted is not important, e.g. webcam streaming)

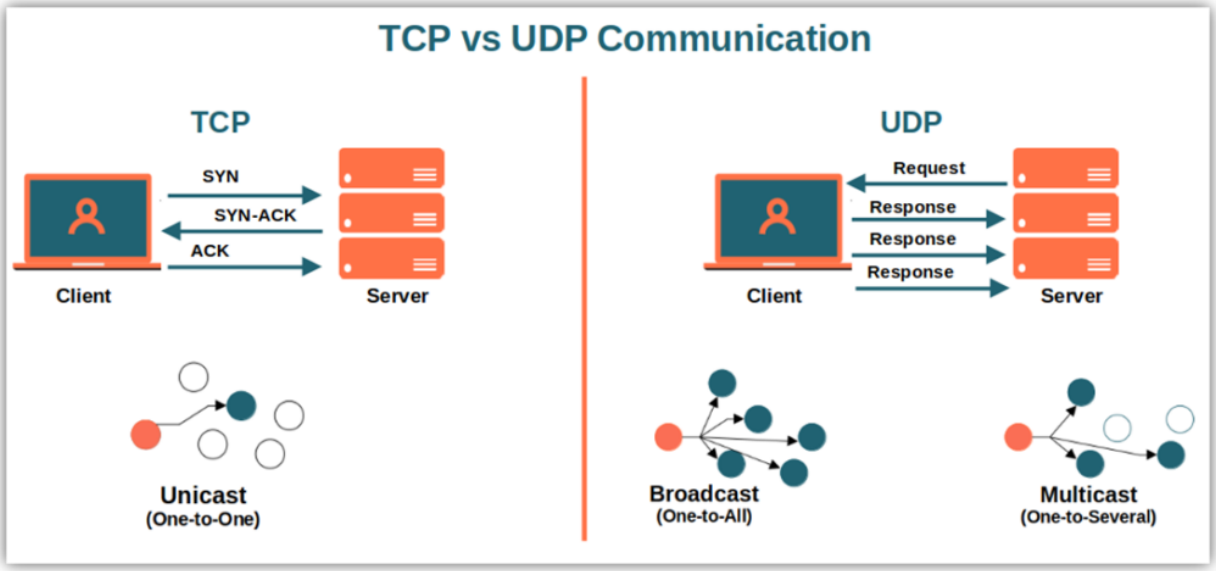


Figure 19. TCP vs UDP in how data is transmitted an the possible types of connections

Multiple computers may be connected in the same network using a plethora of architectures like: bus, ring, mesh, star, tree, hybrid...

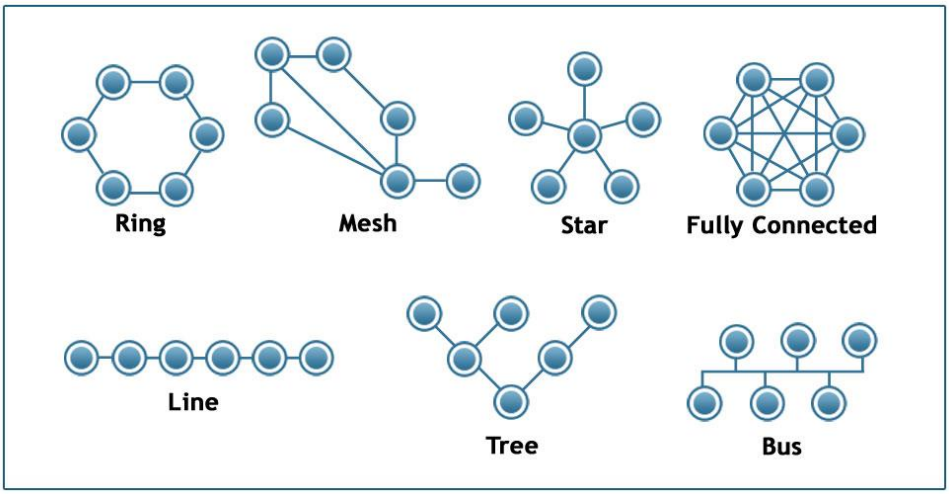


Figure 20. Visual examples of network topologies

1.8 Network services

The Internet network is a computer network where each communication partner has its own unique (public IP) address. Started in 1969 as ARPANET by ARPA (Advanced Research Projects Agency), it provided a decentralized network (destroying one node will not bring down the network) for military usage, and assured only three network services: telnet (remote login), file transfer (by FTP) and remote printing.

Civilian usages started to become apparent with the 1970s to 1980s, when the universities joined it, while operational control was transferred to various entities and it was expanded with CSNET, then while plans were made for decommission, another similar network, the NSFNET funded by National Science Foundation, (NSF) became the backbone of the Internet.

Most known service is WWW (World Wide Web) [22], invented by Tim Berners-Lee in 1989 while working at CERN [21]. The web-pages may be found in the Internet according to a wen-address (URL – Uniform Resource Locator), and are requested by a client (usually a web-browser) from a server (web-server) using the http (Hypertext Transfer Protocol).

Other services include remote connection (telnet, SSH), file transfer protocol (via FTP, SFTP), time services (NTP, SNTP, PTP), directory services (DNS), automatic network address configuration (DHCP), and network management services (ping, traceroute).

A special mention for the DNS service which allows one to access using a website using a name, while the server has in OSI Level 3 an IP address. The DNS resolver does this conversion from name to address as seen in figure below

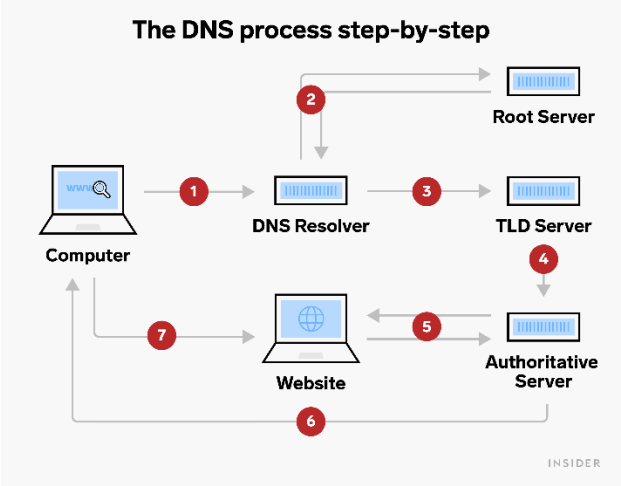


Figure 21. Resolving a website's server address by using DNS hierarchy.

Request 1 is via name and is answered using message 6; request 7 is using public IP address.

1.9 Reaction and feedback

In the world of electronics, reaction is the change seen at the output of a system, as a consequence of a change made at the input of the system.

Feedback refers to the process of returning a part of the signal output back to the input of the system. If the feedback is made in such a way to oppose change (increase of input produces increase of the output which produces the decrease of the input), we will say that negative feedback occurred. If the change on the input produces a change of the output which produces a change *encouraging* the input, then we will say a positive feedback occurred.

Negative feedback loops are useful for making and keeping a system stable.

Positive feedback loops are useful for making oscillators.

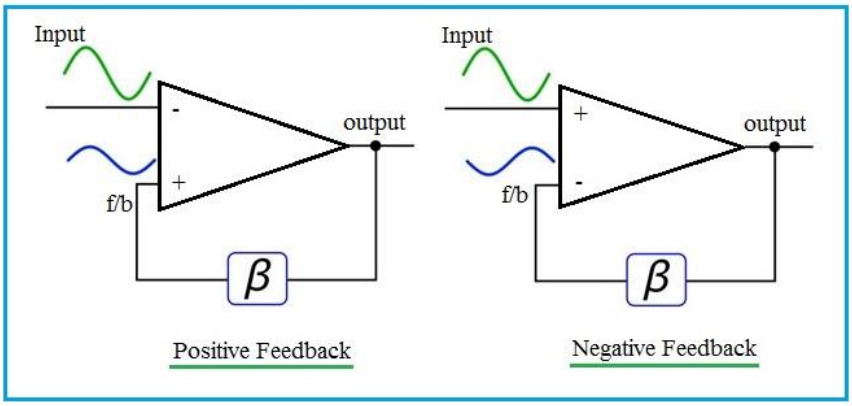


Figure 22. Positive feedback loop (left). Negative feedback loop (right)

Table 3. Comparison between positive and negative feedback in electronics

Parameter	Positive Feedback	Negative Feedback
Input and output voltage, noise	increases due to fb	Decreases due to fb
Fb signal and input signal	in phase	out of phase
Gain	increases	decreases
Stability	poor	better
applications or use	oscillators	amplifiers

The triangle-shaped picture is called “operational amplifier”, is analog circuit block, and is used to output a single-ended voltage (that is, referred to the GND), Vout, as the product of Amplification (A) and input voltage differential (Vdiff = V_+ - V_-). It is used in digital designs

to improve output current capacity in a configuration where A = 1. In analog designs is used in filters or amplifiers. Input impedance is much higher than output imp (Z_{in} >> Z_{out})

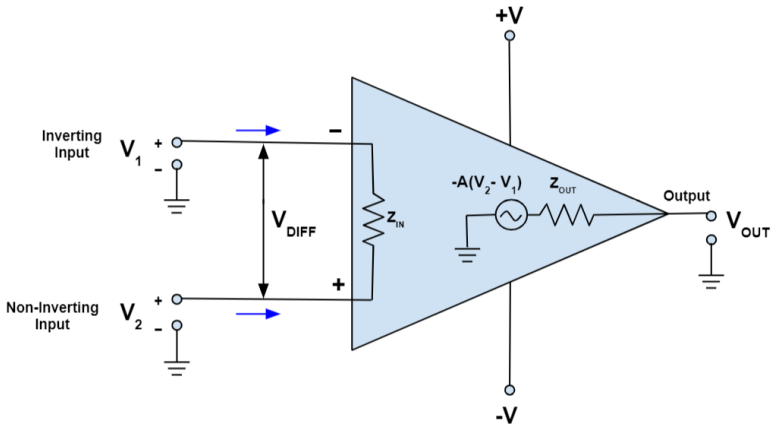


Figure 23. Operational amplifier (opamp)

Just before switching to digital electronics, analog computers were used for calculations. It is obvious how a subtraction could be made using op-amps since in the above example, Vo = A(V2 – V1), but how about multiplication of two voltages? (V2*V1) is usually done by summation of logarithms: logV2 + logV1 = log (V2*V1) and then exponentiation.

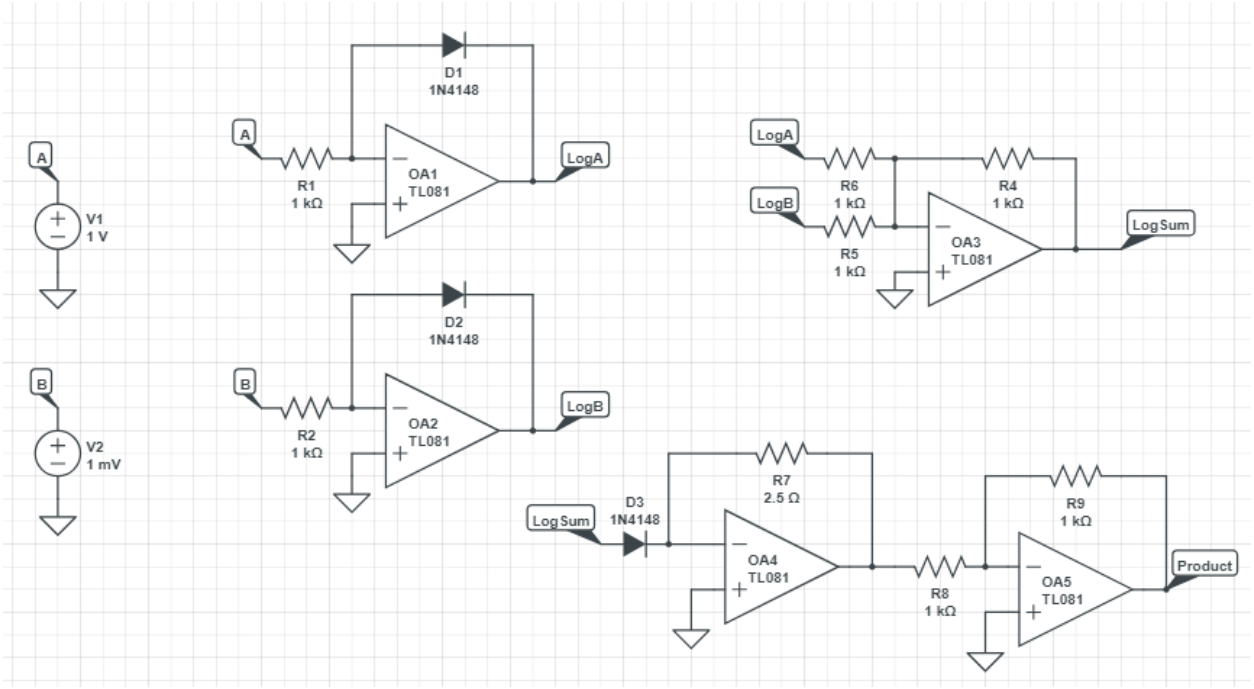


Figure 24. Op-amps configured with diodes or resistors in their feedback loop to do: logarithmation, summation or product

2. Basic hardware bricks

2.1 Binary and Boolean Logic

Boolean algebra is a branch of algebra, where the values of the variables are truth values (true and false) usually coded as 1 and 0 and uses logical operators such as AND (conjunction), OR (disjunction), NOT (negation). It was introduced by English mathematician George Boole in the book "The Mathematical Analysis of Logic" in 1847.

A logical operation is a function of two variables and may be expressed using a truth table as below:

Table 4. A (general) logic operation's truth table

A	B	OP (A, B)
FALSE	FALSE	?
FALSE	TRUE	??
TRUE	FALSE	???
TRUE	TRUE	????

There are 16 dual-input single-output logical operations, 3 of which are most used (hence named). Their truth table is as below:

Table 5. Truth tables for AND, OR and NOT operations

A	B	A and B	A or B	Not A
FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE

Table 6. Truth tables for AND, OR and NOT operations (seen as 1-bit operations)

A	B	A and B	A or B	Not A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

2.2 Digital Gates and Digital Circuits

The electronic circuits used to implement Boolean logic are the logical gates. Nowadays, all gates are made of transistors (a semiconductor device used to amplify or switch electrical signals and power). For example, the NOT gate is made of two CMOS transistors (a p-channel and n-channel MOS transistor), as seen in Figure 4 below.

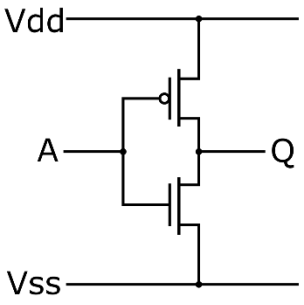


Figure 25. NOT gate ("inverter") made of one pMOS (top) and one nMOS (bottom) transistor.

Vdd is usually at least 1.8V over Vss, and Vss is usually ground. When A is "low," pMOS transistor conducts current and draws Q close to Vdd ("high") whereas nMOS is not conducting. When A is "high" level, nMOS conducts and ties Q to the Vss level ("low").

Engineers use the symbols of such gates, in logic schematics; these symbols, are ratified by international standards as seen in Figure 5.

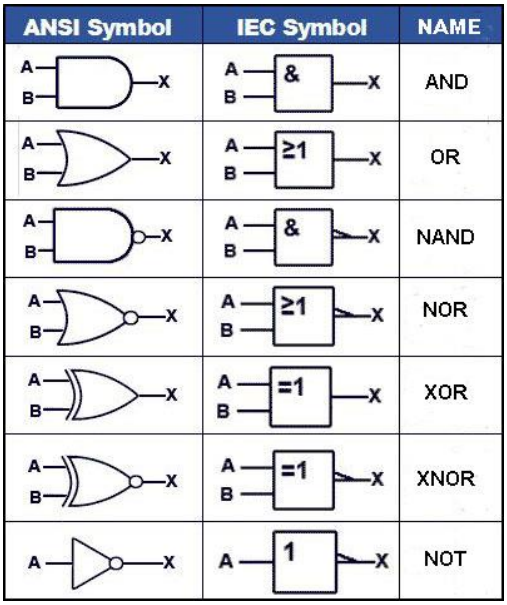


Figure 26. ANSI / IEC [4] (right) and MIL-STD-806B [5] (left) symbols foremost common 7/16 dual-input logic gates (elementary), with their names.

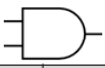
YES

INPUT A	OUTPUT
0	0
1	1

NOT

INPUT A	OUTPUT
0	1
1	0

AND



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

OR

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

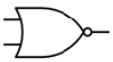
XOR

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

NAND

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

NOR



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

XNOR

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

Figure 27. The truth tables for the most commonly used logic gates.

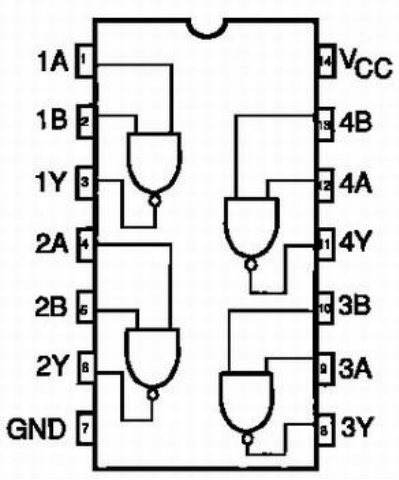


Figure 28. The 74HC00 circuit, a quad-input NAND circuit in DIP14 package.

For SMD it may also come in SOIC14 package. A usual format for quad gates, as the terminals required are: 1 for Vcc, 1 for GND, 4x2 inputs, 4x outputs, therefore 14 pins are needed.

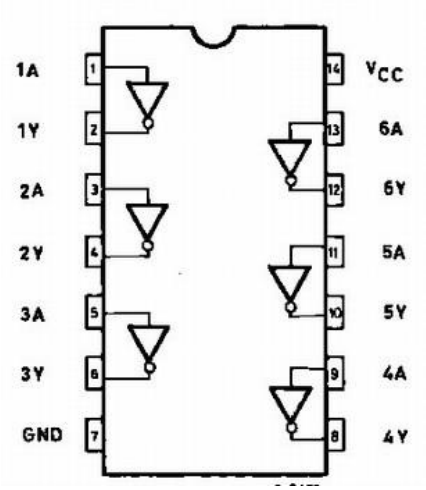


Figure 29. The 74HC04 circuit, a hex-inverter circuit in a 14-pin package (DIP14 or SOIC14).

The pins are as follows: 1 for Vcc, 1 for GND, 6 inputs and 6 output for the 6 NOT gates inside of it

Table 7. A few common 74HC integrated circuits

Circuit name	Description
74HC00	Quad 2-input NAND Gate
74HC02	Quad 2-input NOR Gate
74HC04	Hex Inverter
74HC08	Quad 2-input AND Gate
74HC10	Triple 3-input NAND Gate
74HC11	Triple 3-input AND Gate
74HC14	Hex Inverter Schmitt Trigger
74HC20	Dual 4-input NAND Gate
74HC30	8-input NAND Gate
74HC32	Quad 2-input OR Gate
74HC74	Dual D Flip-Flop
74HC75	Quad BiStable Transparent Latch
74HC85	4-bit Magnitude Comparator
74HC86	Quad EXCLUSIVE-OR Gate
74HC125	Quad Bus Buffer Tri-State
74HC164	8-bit Serial Shift Register
74HC193	Binary Up/Down Counter with Clear
74HC244	Octal Driver Tri-State
74HC595	8-bit Serial-to-Parallel Shift Register Tri-State

Table 8. Common packages for 16-pin IC 74HC595

PART NUMBER	PACKAGE	BODY SIZE (NOM)
SN54HC595FK	LCCC (20)	8.89 mm × 8.89 mm
SN54HC595J	CDIP (16)	21.34 mm × 6.92 mm
SN74HC595N	PDIP (16)	19.31 mm × 6.35 mm
SN74HC595D	SOIC (16)	9.90 mm × 3.90 mm
SN74HC595DW	SOIC (16)	10.30 mm × 7.50 mm

Table 9. Recommended operating conditions for 54HC595 / 74HC595 ICs

			SN54HC595			SN74HC595			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		2	5	6	2	5	6	V
V_{IH}	High-level input voltage	V_{CC} = 2 V	1.5			1.5			V
		V_{CC} = 4.5 V	3.15			3.15			
		V_{CC} = 6 V	4.2			4.2			
V_{IL}	Low-level input voltage	V_{CC} = 2 V			0.5			0.5	V
		V_{CC} = 4.5 V			1.35			1.35	
		V_{CC} = 6 V			1.8			1.8	
V_I	Input voltage		0		V_{CC}	0		V_{CC}	V
V_O	Output voltage		0		V_{CC}	0		V_{CC}	V
$\Delta t/\Delta v$	Input transition rise or fall time(2)	V_{CC} = 2 V			1000			1000	ns
		V_{CC} = 4.5 V			500			500	
		V_{CC} = 6 V			400			400	
T_A	Operating free-air temperature		–55		125	–40		85	°C

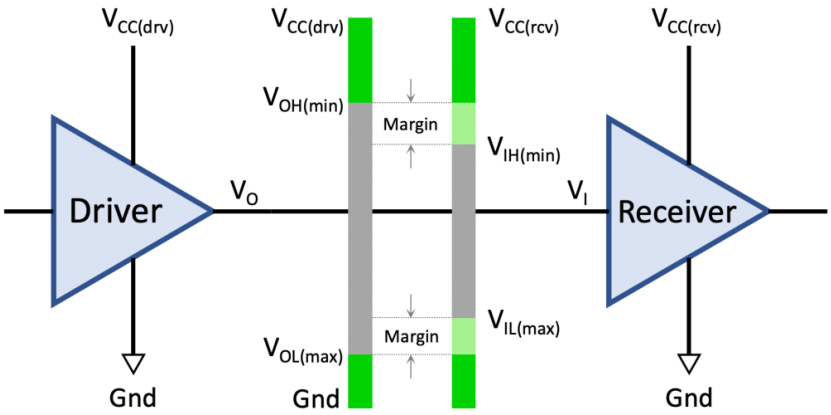


Figure 30. Output requirements vs. input requirements.

- High state/Logic 1: $V_O > V_{OH}$ (V_{OH} is minimum output voltage for state HIGH or logic 1)
- Low state/Logic 0: $V_O < V_{OL}$ (V_{OL} is maximum output voltage for low/0 state)
- High state/Logic 1: $V_I > V_{IH}$ (V_{IH} is minimum voltage interpreted as logic high/1)
- Low state/Logic 0: $V_I < V_{IL}$ (V_{IL} is maximum voltage interpreted as logic low/0)
- "Noise Margin"- allows logic level to be correct even if some voltage variation occurs
- High state/Logic 1: $V_{OH} = V_{IH} + Vnoise$ (noise does not pull voltage below threshold)
- Low state/Logic 0: $V_{OL} = V_{IL} – Vnoise$ (noise does not pull voltage above threshold)

2.3 Analog Elements / Devices / Circuits

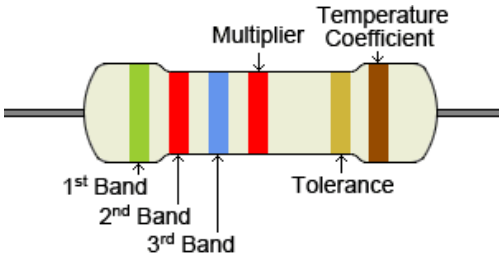
Some devices can only absorb electrical power (they are called passive components) and some devices can both absorb and deliver electrical power (these are called active components).

- Passive components: resistors, capacitors, inductors, transformers
- Active components: semiconductor-based devices (transistors, solar cells, SCRs and most ICs)

2.3.1. Resistors

A resistor is a passive electric component, with two-terminals. They are used to reduce ("resist") the current flow, divided voltages or terminate transmission lines. The main characteristic of this element is its **resistance**, which can be measured as Voltage divided by the current across it, that is Ohm's law: R = U / I

The THT resistors are color-coded, whereas the SMD components are number-coded.



Color	1st, 2nd, 3rd Band	Multiplier	Tolerance	Temperature Coefficient
Black	0	× 1		250 ppm/K(U)
Brown	1	× 10	±1% (F)	100 ppm/K(S)
Red	2	× 100	±2% (G)	50 ppm/K (R)
Orange	3	× 1K	±0.05%(W)	15 ppm/K (P)
Yellow	4	× 10K	±0.02%(P)	25 ppm/K (Q)
Green	5	× 100K	±0.5% (D)	20 ppm/K (Z)
Blue	6	× 1M	±0.25%(C)	10 ppm/K (Z)
Violet	7	× 10M	±0.1% (B)	5 ppm/K (M)
Grey	8	× 100M	±0.01% (L)	1 ppm/K (K)
White	9	× 1G		
Gold		× 0.1	±5% (J)	
Silver		× 0.01	±10% (K)	

Figure 31. Color-coded six-band THT resistor, and the table to decode it. Four bands are for resistance, one for tolerance and one for temperature coefficient.

The above resistor's nominal resistance is: 5 (Green), 2 (Red), Million (Blue), that is: 52 MOhm, with a tolerance of +/-5% (Gold). The actual resistance is therefore between 0.95*52M and 1.05*52M (so between 49.4 MOhm and 54.6 MOhm)

Table 10. Decoding table depending on number of bands painted on THT resistors.

Number of bands	Resistance	Tolerance	Temperature coefficient
3	1,2 with 3 as multiplier	Not specified (default is +/-20%)	Not specified
4	1,2, with 3 as multiplier	4th band	Not specified
5	1,2,3 with 4 as multiplier	5th band	Not specified
6	1,2,3 with 4 as multiplier	5th band	6th band

The ppm stands for parts-per-million, K from Kelvin, and a temperature coefficient of resistance (TCR) is a measure on how much the resistance modifies when temperature modifies (usually we want the resistance to stay fixed regardless on the temperature, but because of thermal noise, the resistance usually increases as electrons flow harder. This effect is intentionally exploited to measure the temperature, in **thermistors,** and depending on the type of t thermistors (PTC-positive temperature coefficient or NTC-negative temperature coefficient) the resistance increases or decreases with temperatures in a predictable way.

TCR = (R2- R1)/ R1 (T2-T1), where R2 and R1 is the value of the resistor's resistance, measured at T2 and T1 temperatures (in Kelvin degrees) respectively. Assuming T2-T1 is 1 Kelvin degree, the TCR means how much the resistance changes at a change of the temperature of 1 degree.

The SMD resistor are not color-coded but they have written alphanumeric characters.

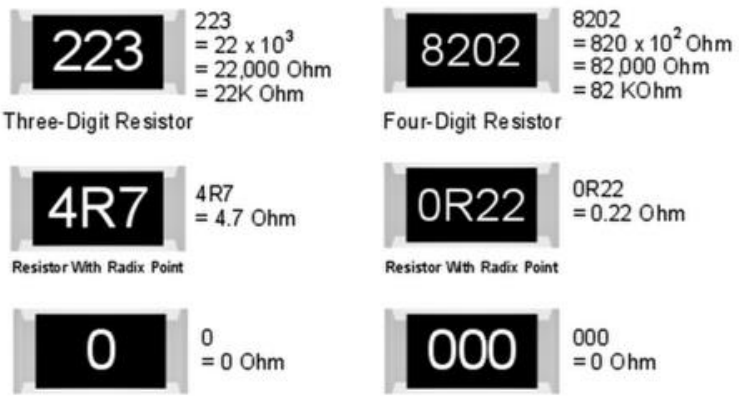


Figure 32. Examples for SMD resistors of various resistance

The "R" letter is the decimal separator for parts of Ohm. 0R22 means 0.22 Ohm, 4R7 means 4.7 Ohm, and when R is missing, the last digit is the multiplier, 8202 is $820*10^2$ Ohms, or 82 kOhm. A particular type of resistor of 0 Ohm is usually used instead of a jumper, for configuring of various circuits (the factory may or may not mount it, depending on the circuit type, in order to reuse a PCB for multiple purposes).

Fixed resistors have a non-tunable resistance, while variable resistors (**potentiometers** or **trimmers**) can be adjusted by hand. A special type of circuit which emulates a resistor (as behavior) is the **digital potentiometer (digipot)** which can vary its resistance depending on the commands it received over a data bus; of course, this is not a passive component.

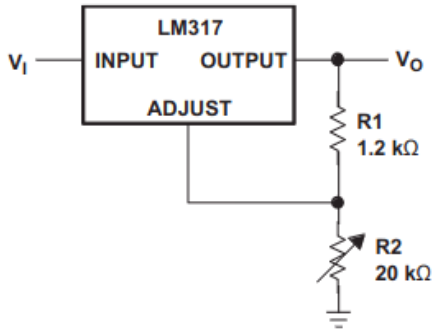


Figure 33. LM317 adjustable regulator configuration via R2 variable resistor

A typical schematic with resistors: the voltage divider, where Vout = [R2 / (R1 + R2)] * Vin. This has the advantages of being able to obtain a smaller voltage from a larger one, at the expense of the power consumption (R1 and R2 dissipate power as heat) and output resistance (a circuit powered by the Vout will not be able to draw current comparable to the one flowing through the voltage divider)

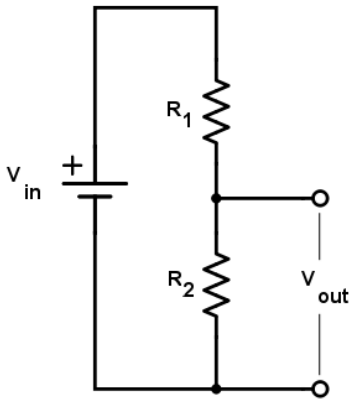


Figure 34. Voltage divider

Another important parameter of the resistor is the maximum dissipated power: the 0805 SMDs usually dissipate about 1/8 Watts (0.125 W) whereas common THT resistors dissipate 0.25W or 0.5W. Special ceramic THT resistors may dissipate even more, and for tens of watts or more, there are resistors fitted from the factory with heat-spreaders.

Figure 35. A 3.3 Ohm THT resistor manufactured to be able to disspate up to 100W

Figure 36. Pictures of different types of resistors

2.3.2. Capacitors

The capacitor is an analog electric device that stores energy as an electric field (by accumulation of electric charges on its two plates). The capacitors are used in filtering applications (they filter the DC signals and allow only AC to pass), DRAM memories for storing information or power-supplies (for power conditioning or for switching in charge-pumps). In large factories, they are used to correct the power factor of large consumers.

The main characteristic of the capacitor is its capacitance, a measure of the amount of energy it may store. It is the proportionality constant between the current through the capacitor and the rate of voltage variation over time. Assuming the voltage is constant, the current is zero. Assuming the current is constant, the voltage increases or decreases (depending on the sign of current) linearly. Similar to the inductor, the capacitor will at some point reach its maximum rated voltage and crossing it will mean the destruction of the insulation between the two plates, and therefore cause an open-circuit.

I = C (dv / dt), where I is current (in Amperes), V is voltage (in Volts), C is capacitance measured in Farads and t is time (in seconds)

Similarly to resistors, there are fixed-value capacitors or variable capacitors.

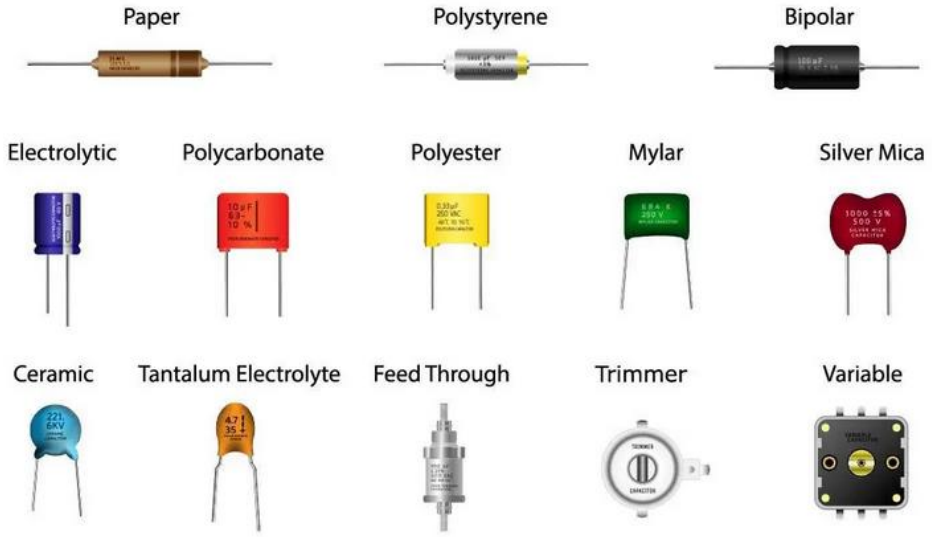


Figure 37. Various types of capacitors

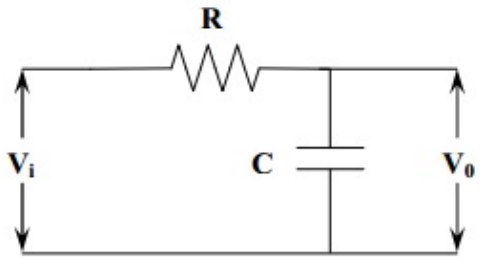


Figure 38. The RC integrator. Vo increases to Vi after infinite time.

An important circuit to be studied is the RC integrator. Assuming Vo is 0 and Vi is applied, the capacitor charges with current Vi / R. After a short time, with the increase of Vo, the charging current decreases: I = (Vi – Vo) / R. The Vo asymptotically gets to Vi after infinite time, however, the capacitor is considered charged after only getting 63% Vi the voltage, which is achieved in RC time.

The RC low-pass filter is a simple low-pass filter with the schematic of RC integrator. The corner (cutoff) frequency is defined as 1/ (2*π*R*C) and represents the frequency where the input power is attenuated by half (or 3dB)

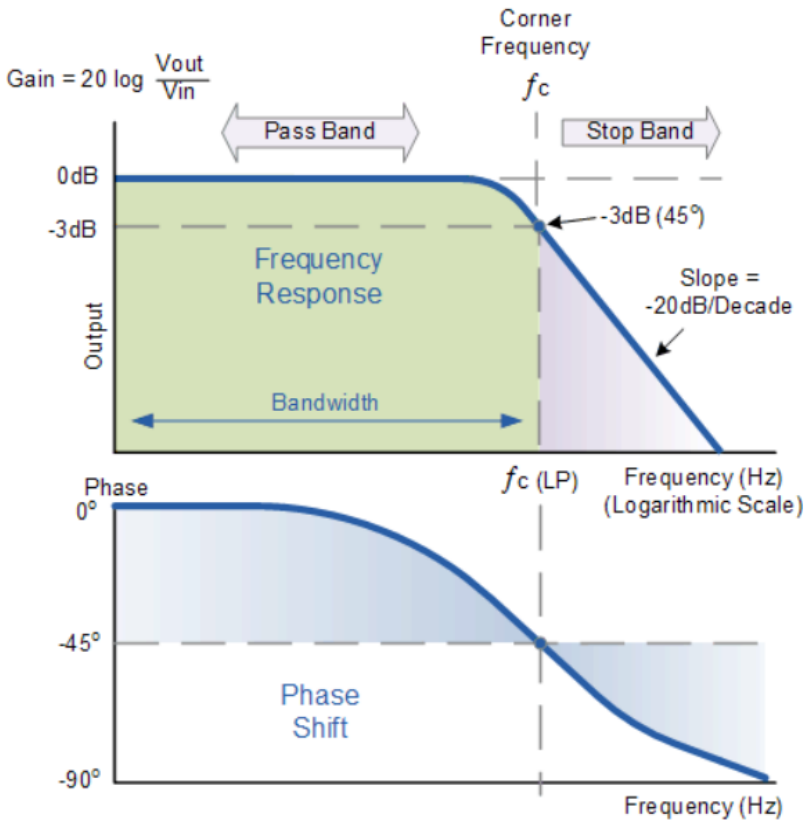


Figure 39. RC low-pass filter Bode plots (for gain and phase)

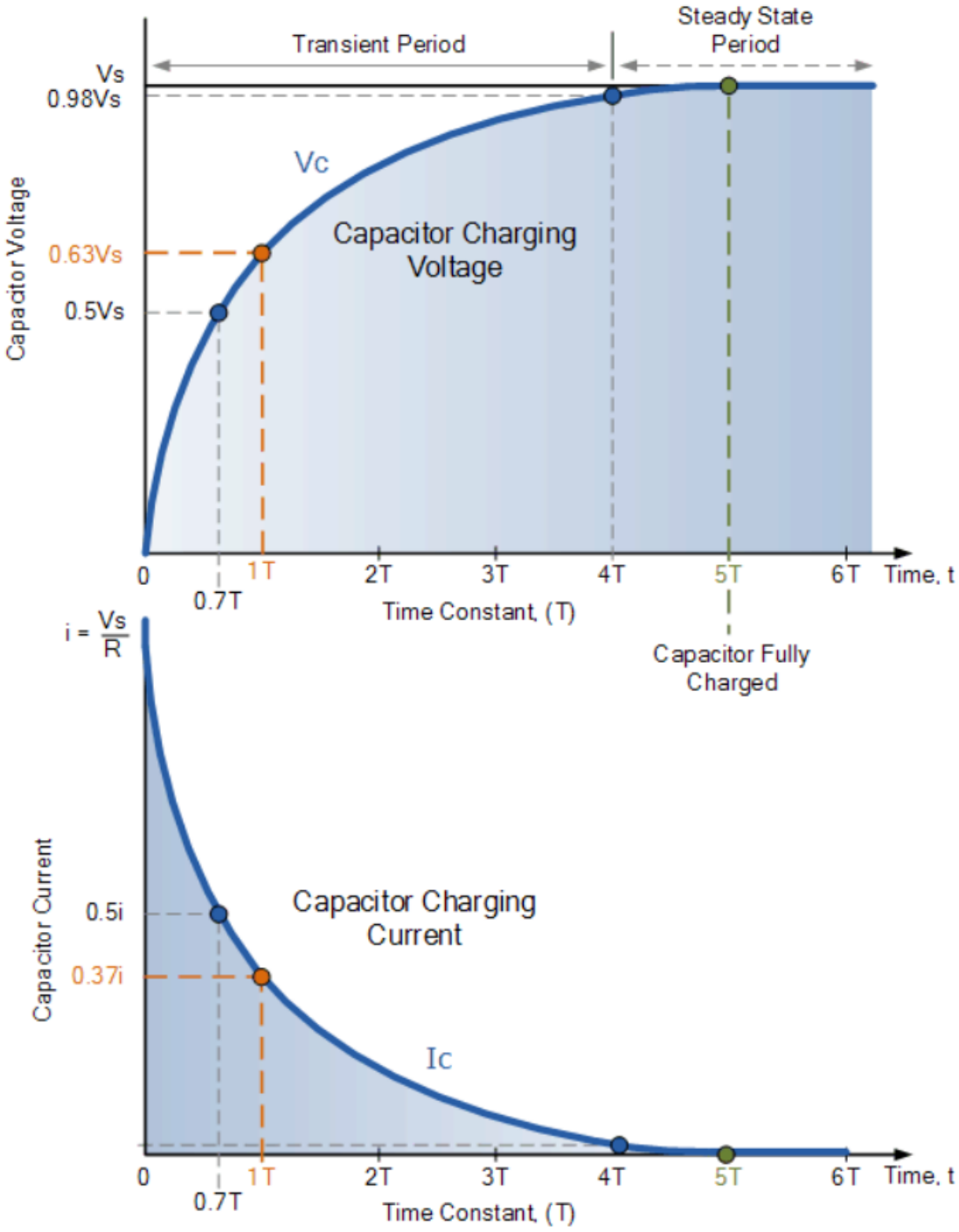


Figure 40. The RC time constant in RC integrator circuit

Charging a capacitor from 0 to 63% of the applied input voltage, will be done in 1*RC time, while charging to 98% will take 4*RC time. Similarly, discharging a capacitor from 100% to 37% (that is 100%-63% = 37%), will be done in 1*RC time.

Table 11. The impact of time in the evolution of capacitor voltage, in RC integrator circuit

Time	Capacitor voltage ratio (charging from 0% Vin)	Capacitor voltage ratio (discharging from 100% Vin)
1*RC	62.8 %	36.8 %
2*RC	86.5 %	13.5 %
3*RC	95 %	5 %
4*RC	98.2 %	1.8 %
5*RC	99.3 %	0.7 %
infinite	100 %	0 %

It is of utmost importance in digital circuits, to understand the way the input capacity charges while powered from constant voltage, as it influences directly the risetime (10% to 90%). Solving for it, we find that time to be 2.2 RC time.

2.3.3. Inductors

Inductors(coils) are passive devices that store energy in the form of magnetic field when the electric current flows through it. While the inductor accumulates/gives back energy, it opposes the transition (self-inductance), following the law:

$$U = -L * d\,Current / d\,time$$

which means the inductance is the proportionality constant between voltage across inductor and change of current through it: it shows how much the voltage increases or decreases, when a change of current occurs in time. If the current is constant, the voltage is zero. If the current increases linearly, the voltage stays constant, but at some point, when reaching a saturation current, the coil will become a low-resistance wire, and create a short-circuit. The main characteristic is the inductance (L), measured in Henry, after the famous American scientist named Joseph Henry (1797-1878) who discovered the effect of self-inductance while building electromagnets (magnets that create magnetic field because of current flowing through them)

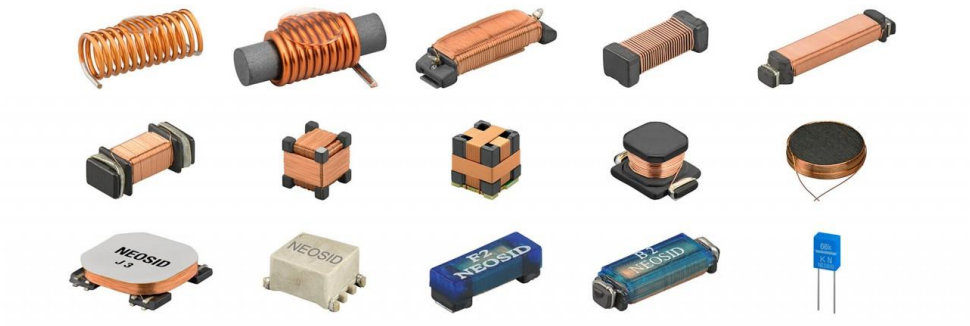


Figure 41. Various types of inductors

2.3.4. Semiconductors

Semiconductors are materials that have the electrical conductivity between the one of a conductor (e.g. copper / metals) and an insulator (e.g. glass, ceramics). The resistivity falls when its temperature rises. The semiconductors can be pure elements (silicon, germanium) or compounds (gallium arsenide, cadmium selenide). They are used in all electronic devices (by the usage of following components: diodes, transistors, integrated circuits)

2.3.4.1 Diodes

Diodes are used mostly in applications for rectifying voltage, protection, switches, signal modulators, signal mixers and oscillators. Its main property is to conduct electric current in one direction only. The main characteristic of the diodes is the U/I plot.

Figure 42. Various diodes formats and types

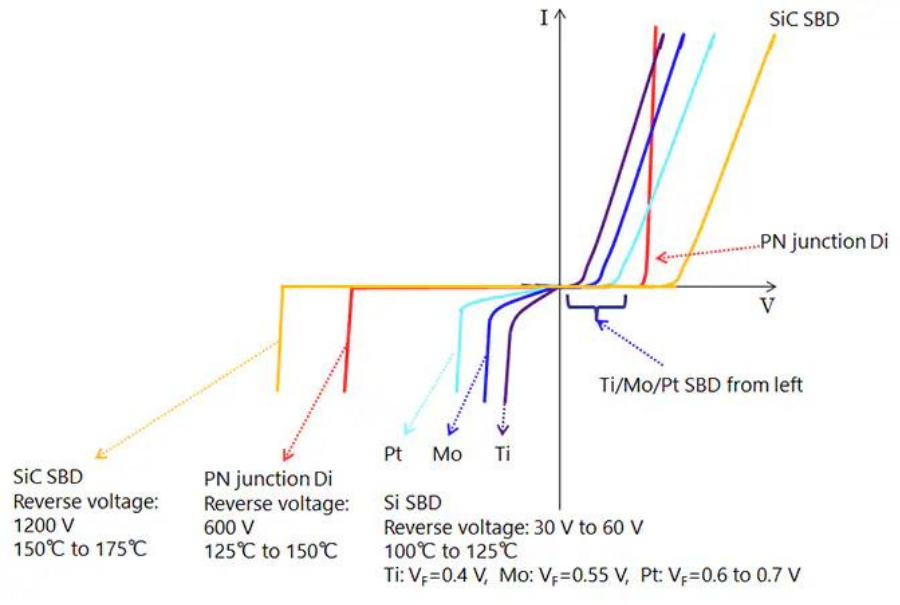


Figure 43. U/I plot for various diodes with their reverse voltages and maximum temperature

PN-junction diode is the most common type of diode and is formed by fusion of a p-type semiconductor (one that has missing electrons) with a n-type semiconductor (one that has extra electrons) to create a potential barrier voltage across the diode junction. Until a certain voltage (e.g., 700mV if silicon-based), the diode will not conduct as the barrier is not overcome, however, adding more voltage across it, will make the diode conduct.

Schottky diodes are notorious for their low forward-voltage (100-400 mV) whereas usual silicon-based diodes have the forward-voltage at around 700mV.

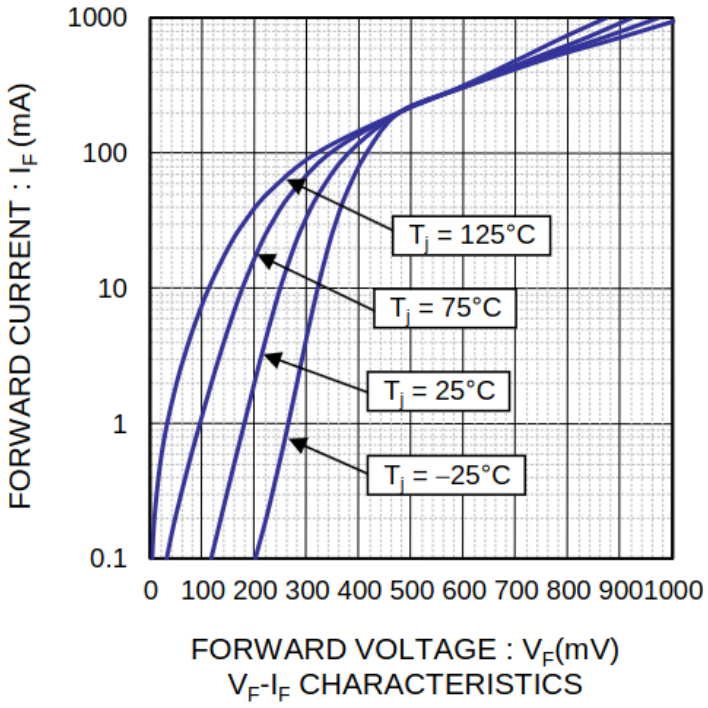


Figure 44. The current as a function of forward voltage and temperature in RB451UM Schottky diode

Another particular type of diode are the **LED (light-emitting diode)**. LEDs are heavily used in displays, lightbulbs, or indicators. The color (given by the emitted wavelength) of the LED will depend on its chemistry. In addition there are RGB LEDs which contain three LEDs (R,G,B) that are spatially close, and can be toggled on or off, independently:

Table 12. LED color depending on wavelength and chemistry for Kingbright KPA-2107 / KA-3021SYCKT /F3** LED*

Color	Wavelength (nm)	Chemistry
Green	525	InGaN
Red	625	AlGaInP
Blue	470	InGaN
Yellow*	590	AlGaInP
Infrared**	940 / 880	GaAs / GaAlAs

Infrared LEDs are used in remote controls (air-conditioning, TVs) while ultraviolet LEDs are used in sunbathing or manufacture of low-cost PCBs (UV presensitized copper clad boards)

In order to differentiate IR remote control commands from the sunlight (sun also emits infrared light), the IR signal is modulated (OOK modulation with carrier of 36 to 40 kHz). While modulation is easy to be performed by a microcontroller (for 40 kHz, it toggles the LED pin every 25 ms), the demodulation is easier to do using a dedicated IC, like the ROHM's RPM7140.

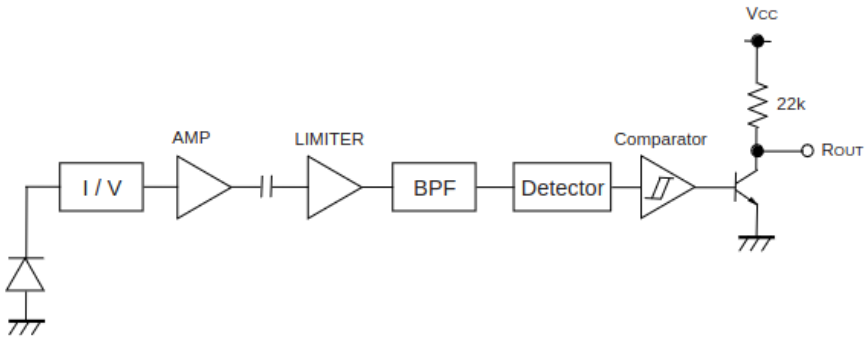


Figure 45. The RPM7140 infrared receiver: internal block schematic: amplifier, limiter, band-pass filter, detector and comparator. Output is open collector with 22kOhm integrated pull-up resistor

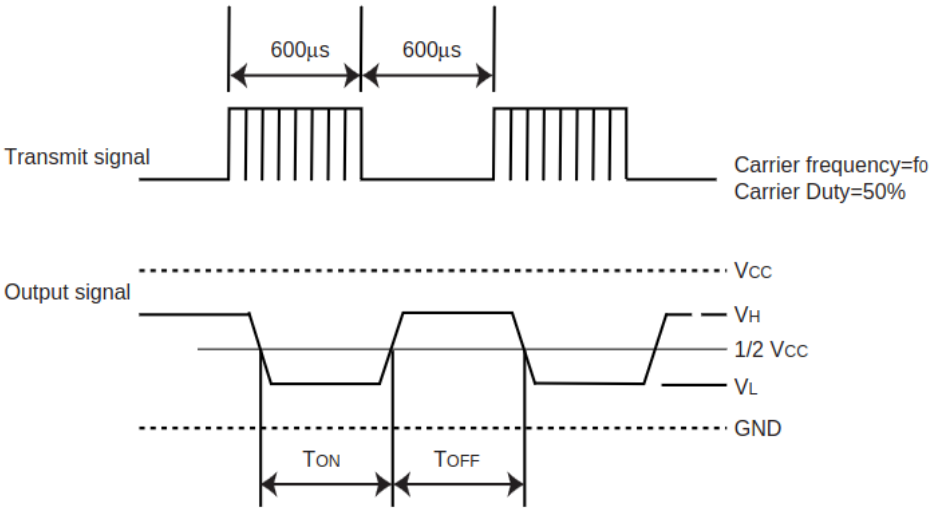


Figure 46. The RPM7140's output signal (down) as reaction to input light (up).

Color temperature is a measure of the color of light, which would describe it in comparison with the emitted light of an idealized non-reflective, opaque body. Note that although it is measured in Kelvin, it does not describe the actual temperature of the body. It is very important for photographers, that need to be careful with it, as the digital cameras' sensor

is not similar to the human eye, therefore improper color temperature settings will cause human skin to be far from the expected pink.

Table 13. Types of white light (by color temperature)

Type of white color	Temperature of color (Kelvin)	Recommendations
Warm white	2700	Calm and relaxing light, great for bedrooms, living rooms…
Natural white	4000	It closely mimics sunlight
Daylight white	5500	Crisp and clear light, great for garages, offices
Cool white	6500	Commercial and industrial applications

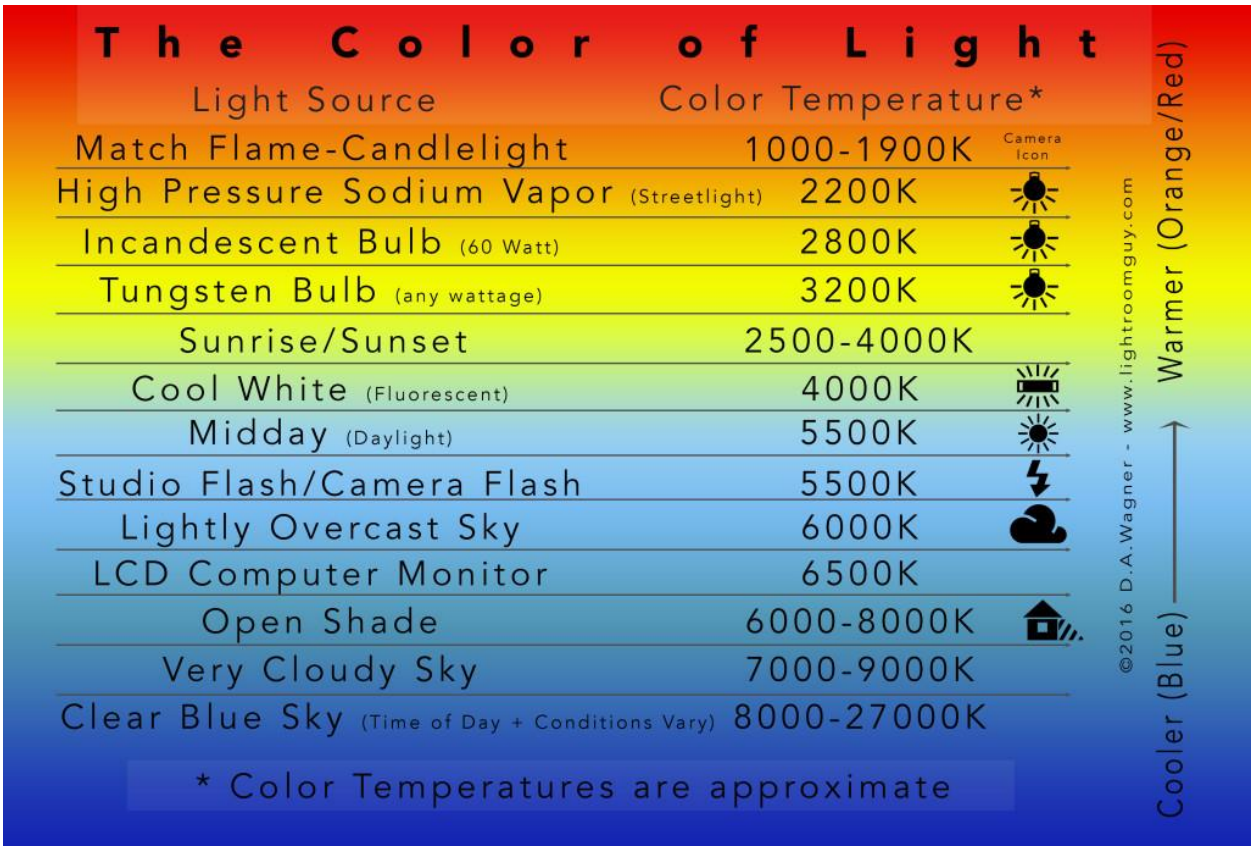


Figure 47. Various color temperatures and their equivalence in real-world

Zener diodes

The Zener diodes use the breakdown region to keep the reverse voltage almost constant (large current increase/decrease produce small voltage increase/decrease).

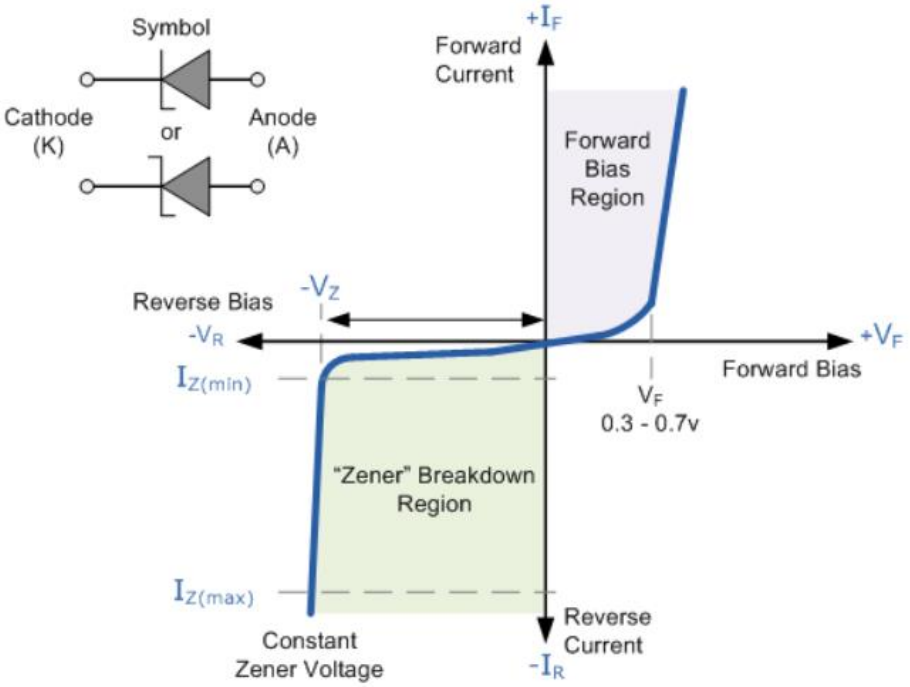


Figure 48. Zener diode's operating regions

Transient voltage suppressors (TVS)

The TVS absorbs very high overvoltage (kilovolts) in a short amount of time to protect downstream electronics.

2.3.4.2 Transistors

Transistors are semiconductors that regulate the current flow (if used as on/off switch, as in digital electronics) or amplify the voltage or current (they increase the strength of a weak input signal and output it for the next stage). The first type of transistor (the point-contact transistor) was invented in 1947 by John Bardeen, Walter Brattain and William Schockley at AT&T Bell Labs, for which they received the Nobel Prize in Physics, in 1956, for their research on semiconductors and their discovery of the transistor effect. The first use of transistors was in telecommunications equipment for reading the mechanical encoding from punched metal cars. The first pocket transistor radio was demonstrated in 1952, whereas the first mass-produces transistor radio was released in 1957 by Sony. Transistors are key active components in all modern electronics and therefore, they are considered one of the greatest inventions of 20th century.

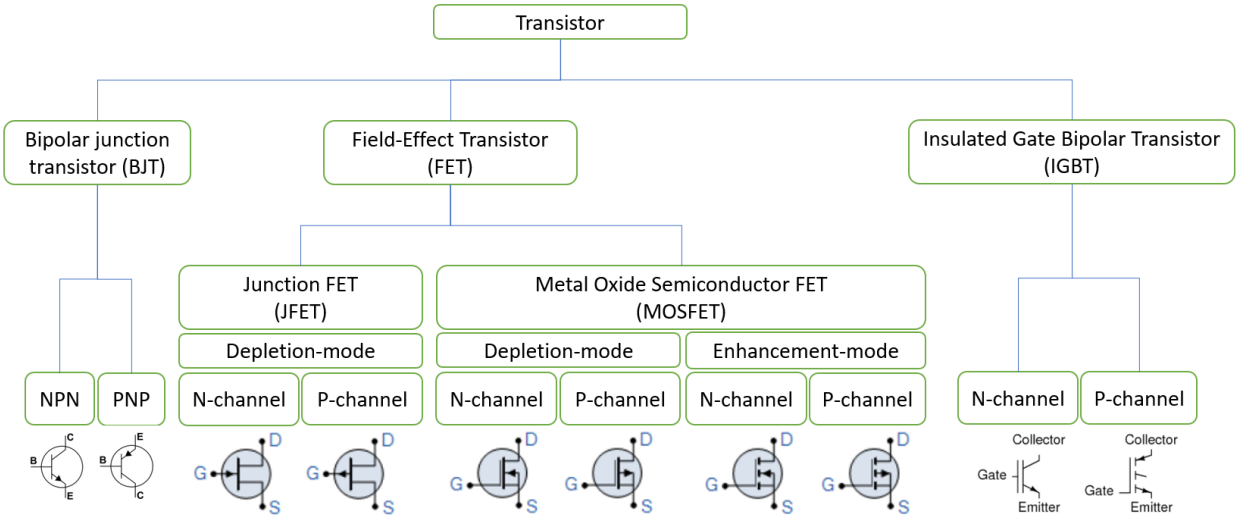


Figure 49. Types of transistors

BJT transistors, although old, are still used today in low-cost schematic, where the 10-40x amplification is enough, or in high-speed switching applications where the FET transistors exhibit large switching losses.

Table 14. Regions of operation for BJT transistors

Voltage relations	**NPN**	**PNP**
VE < VB < VC	Active	Reverse
VE < VB > VC	Saturation	Cutoff
VE > VB < VC	Cutoff	Saturation
VE > VB > VC	Reverse	Active

Table 15. Explanation of BJT regions

Region name	**Interpretation**
Saturation	transistor acts like short circuit: current flows freely from collector to emitter (“on mode”)
Cut-off	transistor acts like open-circuit. No current flows from collector to emitter
Active	the current from collector to emitter, is proportional to the current flowing into the base. Transistor amplifies the base current with a B factor, setting the collector current
Reverse-active	the current from emitter to collector is proportional to the current flowing into the base

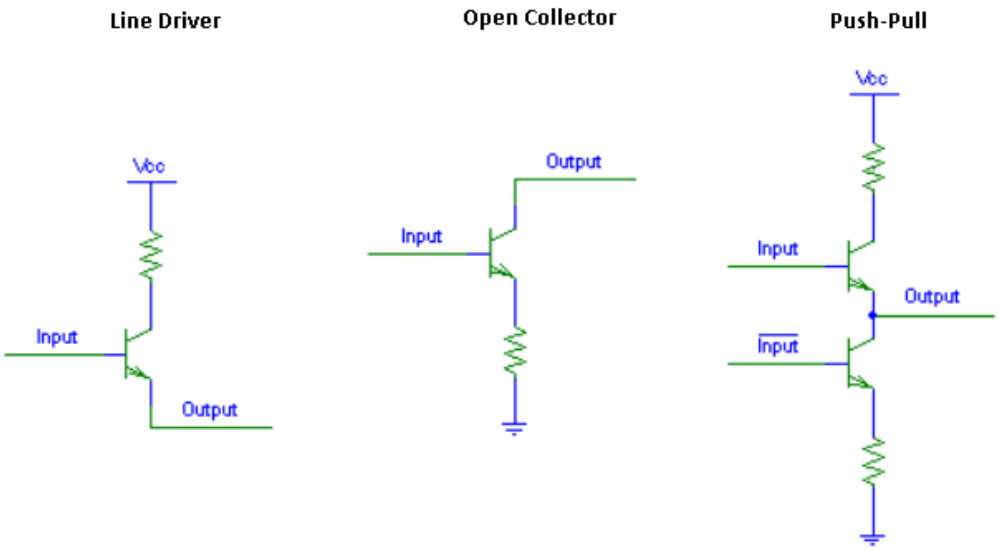


Figure 50. Common usages for BJT: line driver, open-collector and push-pull

The line driver is a sourcing output: when in on state, out will supply Vcc; when in off state, output will float.

The open collector is a sinking output (drives hard to ground). In the ON state, it will supply a path to ground. When in OFF state, output will float.

The push-pull output is a combination of line driver and output collector: because of this, both a source to Vcc or sink to GND is supplied, depending on the input. It is an inverter, as the output is the opposite to the input. (HIGH->LOW, or LOW->HIGH)

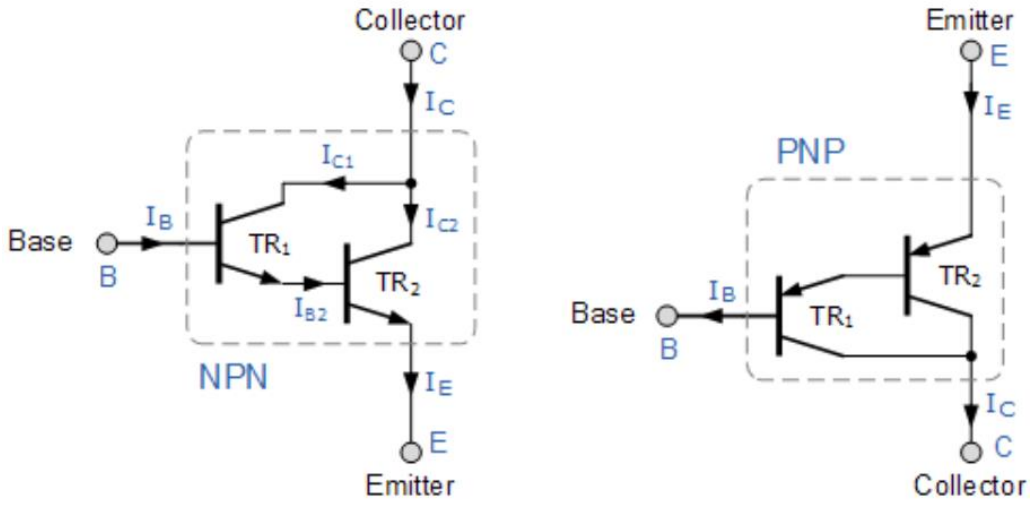


Figure 51. Darlington transistors. The TR1 and TR2's amplification are multiplied

FET transistors although have larger input capacitance (and are harder to drive), they exhibit very high input impedance (virtual no current draw into the gate, after input capacitance is loaded), have very low output resistance when in ON state (RDS-on – can be as little as 1 milliohm), and can drive very large drain currents (hundreds of Amps)

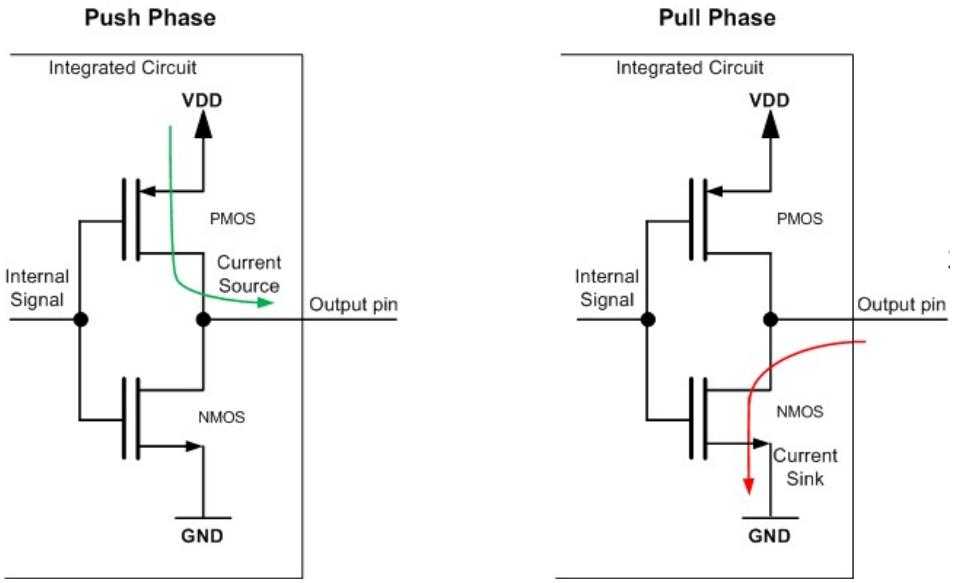


Figure 52. Push-pull stage with FET transistors, usually found in output pads of microcontrollers

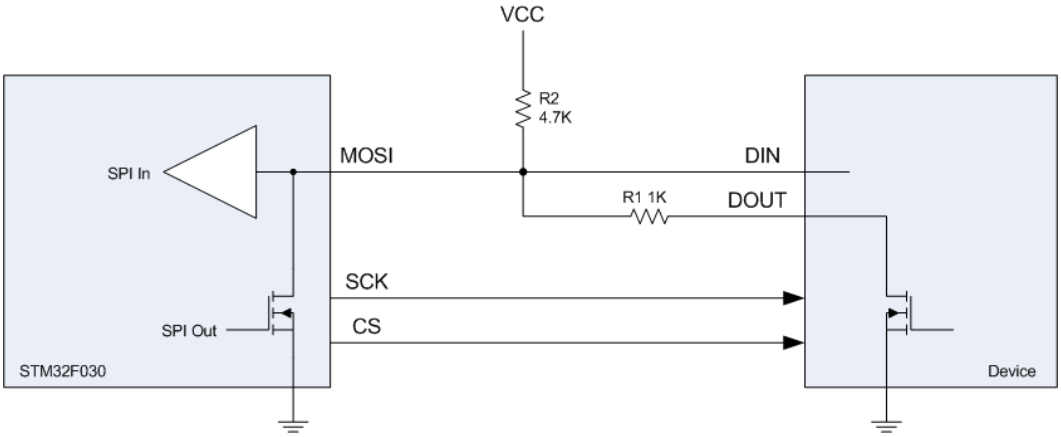


Figure 53.Half-duplex communication between STM32 and another device

Having open drain connections, both STM32 and Device can send data from one to the other, and can even send different data in same time (but the device putting a 0 will win)

2.3.5 Switches

Switches are electrical devices used to make or break an electrical circuit. The following terminology is used to express types of switches:

NO = normal open (the switch stays default in OFF position)
NC = normal closed (the switch stays default in ON position)
Poles = (number of) the circuits that is controlled by the switch
Throws = (number of) the positions that the switch can adopt
SPST = single pole, single throw
SPDT = single pole, double throw

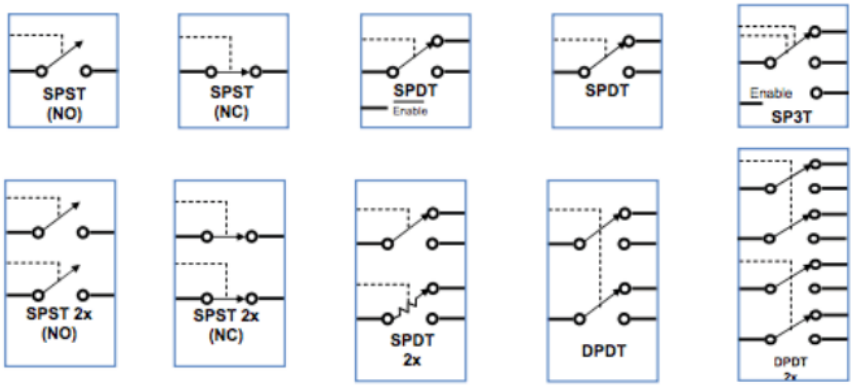


Figure 54. Types of switches

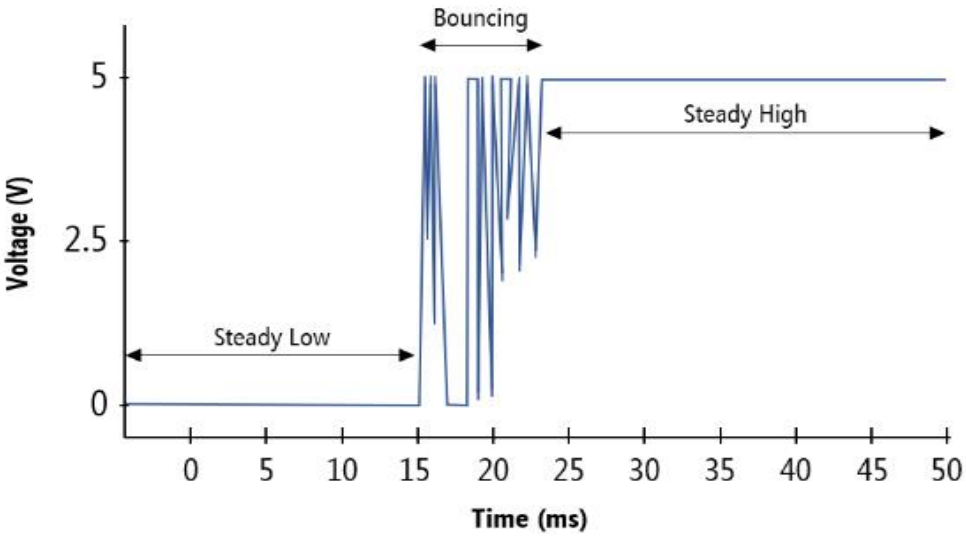


Figure 55. Contact bouncing problem

When the mechanical switch starts to conduct current (goes from OFF to ON), the mechanical contact is imperfect for a small amount of time: this may cause problems if the next system tries to read whether the button was pressed once or twice. Solving this problem is called "debouncing" and can be made either in analog (using a RC filter), or in digital

(software that delays reading on the switch state for at least e.g. 10 ms from toggling, such that the bouncing ended). Below there is a catalog snapshot of CUI Devices switches:

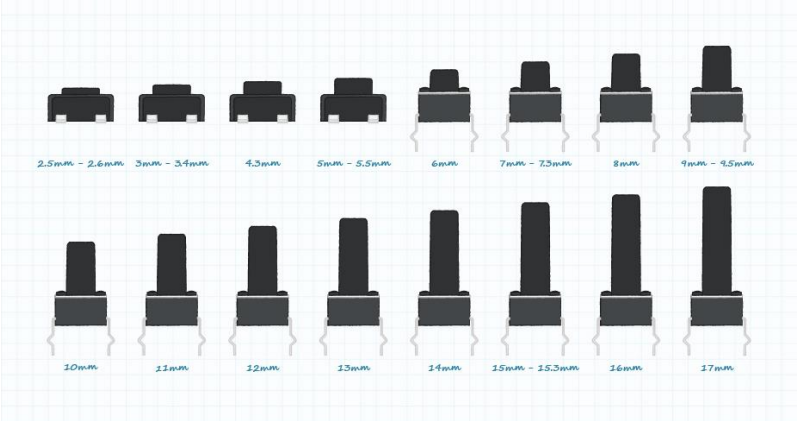


Figure 56. Tactile switches, SMD (top left) or THT

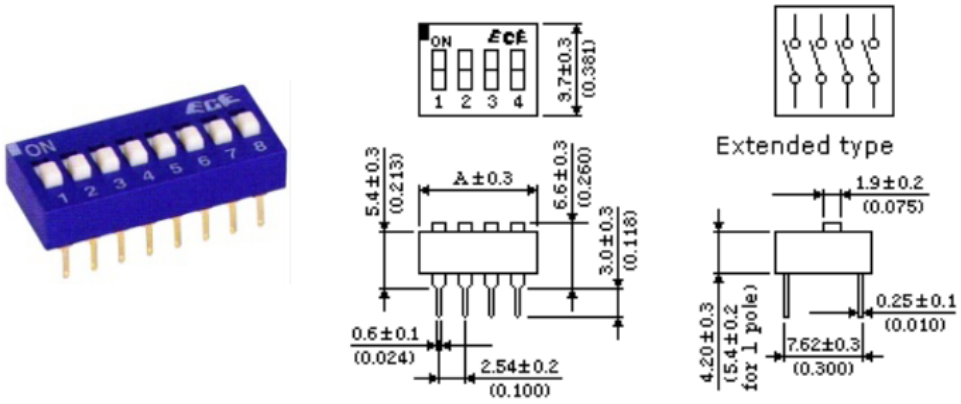


Figure 57. Example of 8-slot DIP-switch

Usually, DIP-switches are used for configuring ICs, by shorting or not an I/O line of the IC.

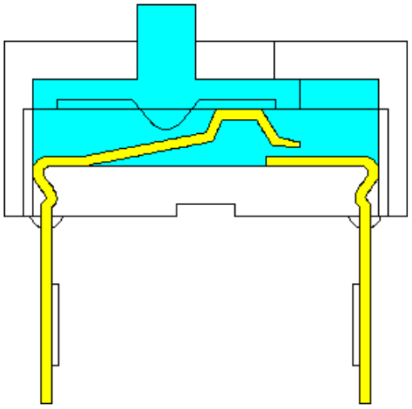


Figure 58. Internal construction of a DIP switch

2.4 Prerequisites for typical analog applications

2.4.1 Prerequisites: the decibel (dB)

The decibel (dB) is a relative unit of measurement, expressing a ratio of two values of power quantity (energy density, acoustic intensity, luminous intensity) or root-power quantity (voltage, current, sound pressure, electric field strength, speed, charge density) on a logarithmic scale. The logarithmic scale is advantages when the range to represent is very large (very small values to very large values). The decibel is 1/10 of a Bel.

The A-weighted decibel (dBA) is an expression of relative sound loudness, as perceived by the human ear. It gives more value to frequencies in the middle of the human hearing an less value to the frequencies at the edges (see ISO 226:2003 standard). Human ear can hear sounds from 20Hz to 20 kHz but is more sensitive in the range of 250 – 5 KHz.

According to National Institute on Deafness and Other Communication Disorders (NIH), there are multiple steps for the human to hear exterior sounds [23]:

- Sound waves enter the outer ear and travel through a narrow passageway called the ear canal, which leads to the eardrum.
- The eardrum vibrates from the incoming sound waves and sends these vibrations to three tiny bones in the middle ear. These bones are called malleus, incus, and stapes.
- The bones in the middle ear amplify, or increase, the sound vibrations and send them to the cochlea, a snail-shaped structure filled with fluid, in the inner ear. An elastic partition runs from the beginning to the end of the cochlea, splitting it into an upper and lower part. This partition is called the basilar membrane because it serves as the base, or ground floor, on which key hearing structures sit.
- Once the vibrations cause the fluid inside the cochlea to ripple, a traveling wave forms along the basilar membrane. Hair cells—sensory cells sitting on top of the basilar membrane—ride the wave. Hair cells near the wide end of the snail-shaped cochlea detect higher-pitched sounds, such as an infant crying. Those closer to the center detect lower-pitched sounds, such as a large dog barking.
- As the hair cells move up and down, microscopic hair-like projections (known as stereocilia) that perch on top of the hair cells bump against an overlying structure and bend. Bending causes pore-like channels, which are at the tips of the stereocilia, to open. When that happens, chemicals rush into the cells, creating an electrical signal.
- The auditory nerve carries this electrical signal to the brain, which turns it into a sound that we recognize and understand.

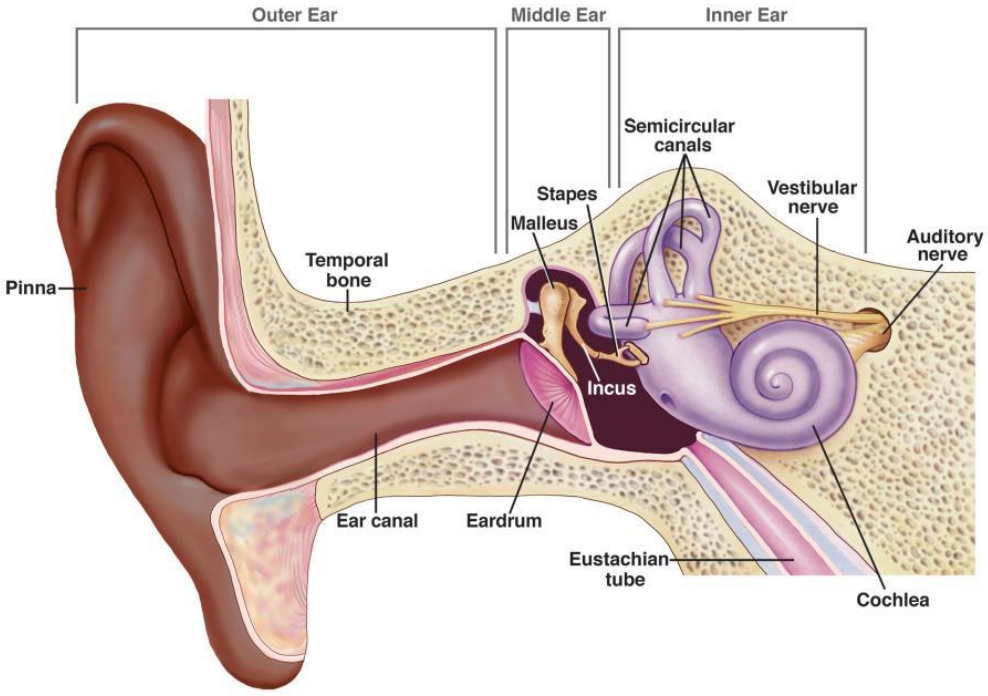


Figure 59. Hearing for humans (source: NIH/NIDCD)

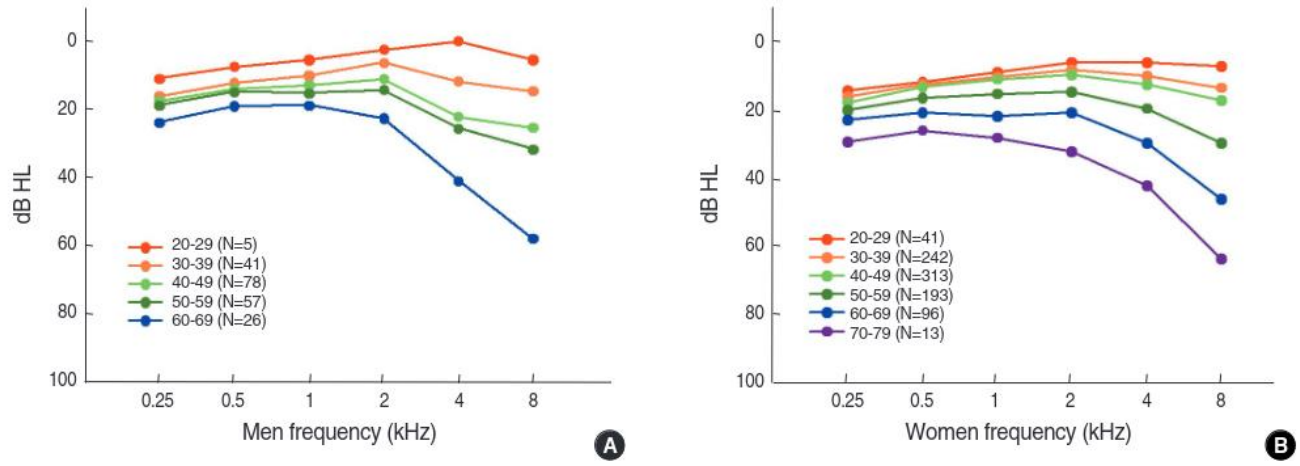


Figure 60. Hearing loss in humans, depending on sex and age

In the figure above, paper [25] did a study regarding the hearing loss on 3470 Korean people (1947 men and 1523 women). Significant differences were found at 4 and 8 kHz where the hearing loss was worse in men than in women.

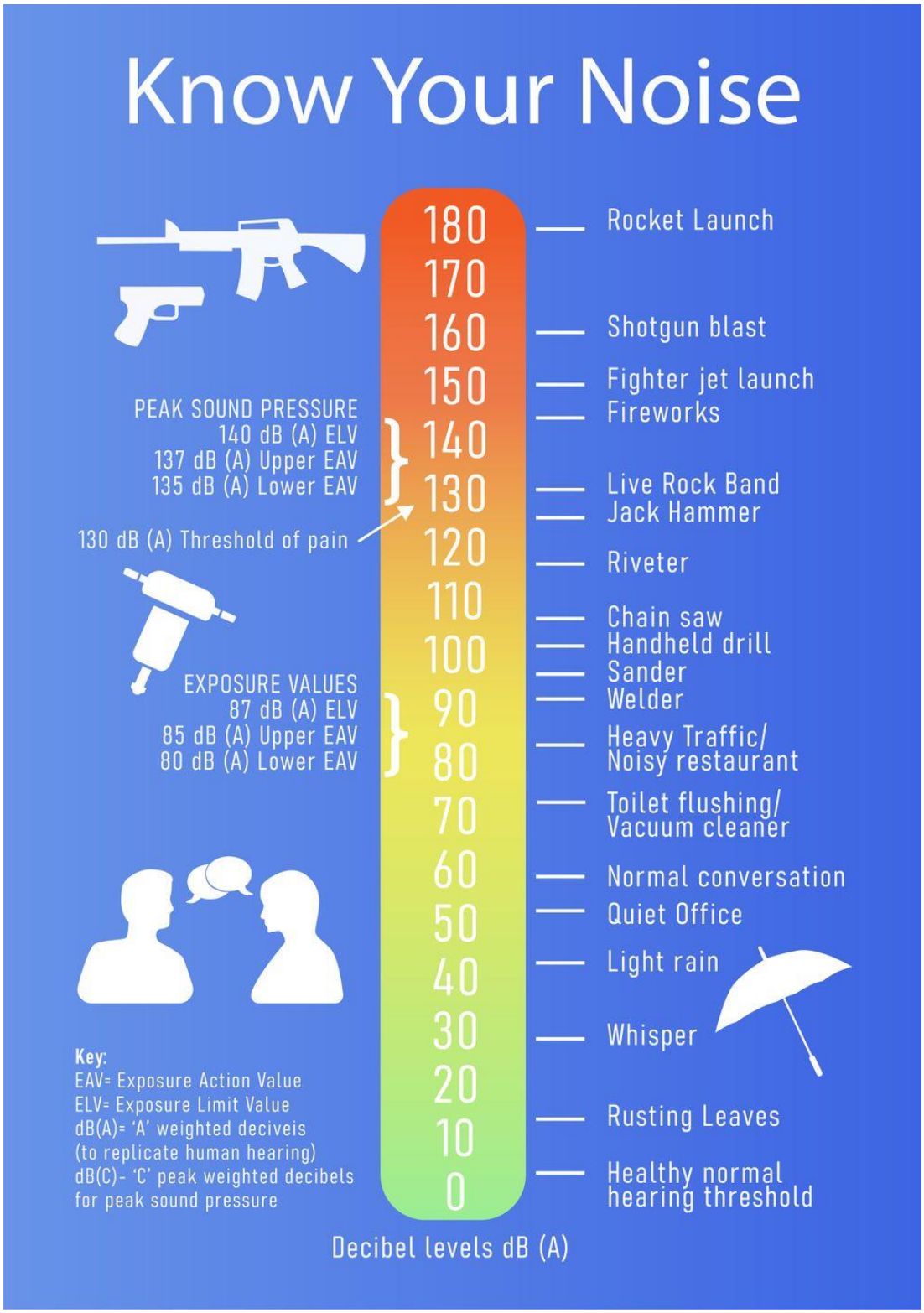


Figure 61. Examples of noises a human may perceive

The dBm is a unit of level used to express a power level, relative to a power of 1 milliwatt.

$$X = 10 \lg (P / 1mW), \text{ where lg is } \log_{10}$$

For expressing ranges from $3.6 * 10^{49}$ W (black hole collision, power radiated in gravitational waves following the collision GW150914), to 0.178 femtowatt (received signal power of a GPU satellite), one uses easily the 526 dBm to -127.5 dBm which is much easier to understand, express and compare.

The dBm is used to express maximum radiated power in FM radio stations, or VHF/UHF receivers, or 2G / 3G / 4G cellphones, minimum received power to maintain a certain speed in Wi-Fi networks etc.

In the below figure, paper [26], the authors built a table of power ratio and voltage ratio vs dB level. As in RC filter before, the -3dB is 0.5 power ratio.

	Power ratio	Voltage ratio
−20 dB	0.01	0.1
−10 dB	0.1	0.32
−3 dB	0.50	0.71
−1 dB	0.74	0.89
0 dB	1	1
1 dB	1.26	1.12
3 dB	2.00	1.41
10 dB	10	3.16
20 dB	100	10
$n \cdot 10$ dB	10^n	$10^{n/2}$

Figure 62. Overview of dB key values and their conversion into power and voltage ratios

Table 16. Other types of decibels (dB)

Type of decibel	Reference value	Notes
dBSPL		For measuring sound pressure, 0 dBSPL is the quietest sound a human may hear
dBu	0.775 V	For measuring voltages
dBv	1 V	For measuring voltages
dBVU	0.775 V	For measuring volume
dBFS	Full scale	0 dBFS is the maximum number a particular digital system can represent

2.4.2 Prerequisites: Fourier transform and Fourier series

The Fourier transform is a mathematical technique which transforms a function of time x(t) to a function of frequency X(w). It is closely related to Fourier Series (expansion of a periodic function into a sum of sin and cos functions).

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(n\,x) + \sum_{n=1}^{\infty} b_n \sin(n\,x)$$

Figure 63. The Fourier series of a function f(x)

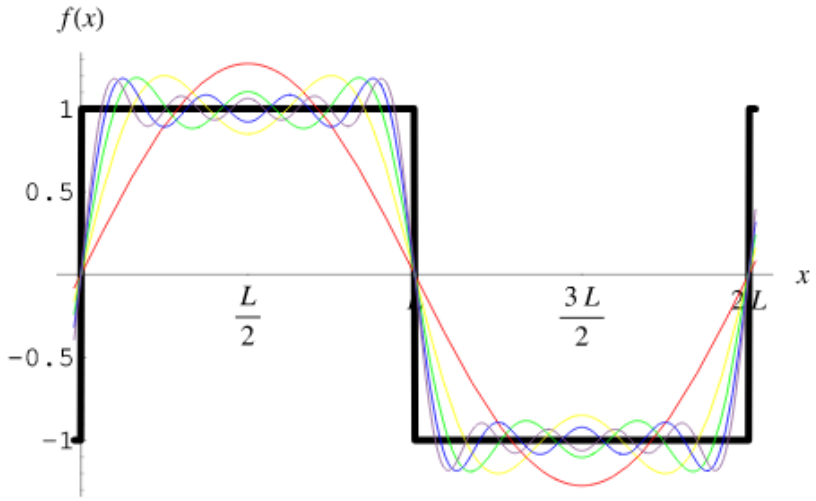


Figure 64. Decomposition of a square wave into first four terms of Fourier series

Note that the continuous component (the horizontal part called **level**, is not a problem since it does not change, but the vertical part of the signal, called edge, represents a big problem for all circuits since in changes state from bottom to top in 0 time – this in fact means the input capacitance of the next circuit, which is not zero, gets charged in zero time, which means the circuit creating the wave gives infinite power at its output). As a consequence, real edges are not 0 in time, but have a risetime (in range of picoseconds or nanoseconds, depending on the circuit type)

There are multiple Fourier transforms depending on the type of the function and the processing time used in conversion:

- CTFT (continuous time Fourier transform)
- STFT (short-time Fourier transform)

Bottom line: the Fourier transform gives a frequency-based representation of the time-based function.

2.4.3 Prerequisites: the Doppler effect

The Doppler effect is the apparent change in frequency of a wave in relation to an observer moving relative to the wave source. It was named after the physicist Christian Doppler, who described the phenomenon in a paper in 1842.

$$f_o = \frac{v + v_o}{v + v_s} f_s$$

Figure 65. Formula for observed frequency (in Doppler effect)

Where fo is the observer frequency of sound, v is the speed of sound waves (340 m/s in air), v_0 is the observer velocity, v_s is the source velocity and f_s is the actual frequency of sound waves.

The effect is usually heard when an ambulance approaches you (and the siren seems having higher frequency), and after passing you, it seems to drop in pitch (lower frequency siren). This is the effect used in Doppler RADARs (Radio Detection and Ranging) by the Police.

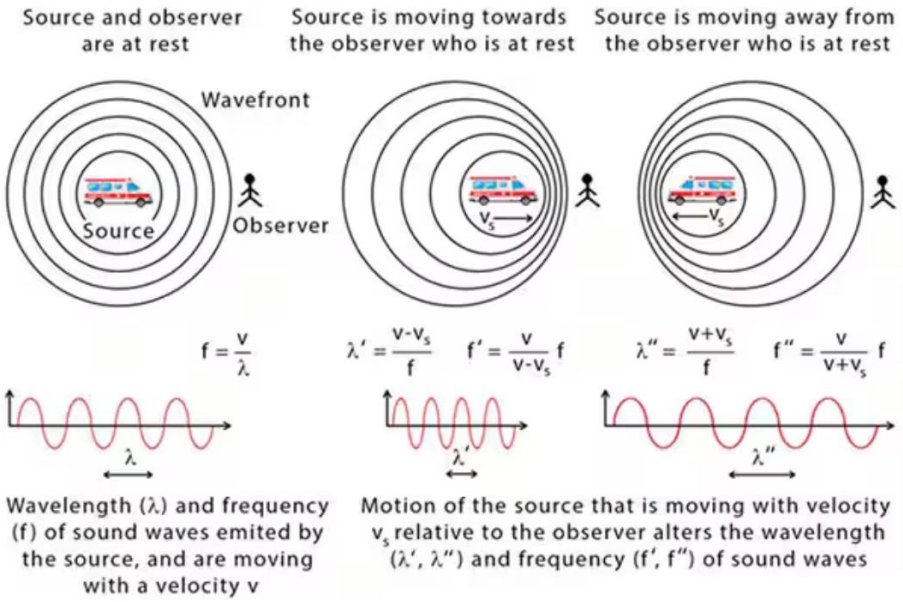


Figure 66. Doppler effect: as the source and observer move closer, the distance between successive wave crests decreases, resulting in a rise in perceived frequency; the opposite is the case as the two move apart. (Science Facts)

The LIDAR (light detection and ranging) uses light instead of sound, or microwaves to detect range. It is sent by a transmitter, is reflected from the ranged-object, and it is received back. The difference in time from the transmission to reception is the space times speed of light.

2.4.4 Prerequisites: the electromagnetic waves

Electromagnetic waves are a form of radiation that travel through the universe. They are formed when an electric field (red) couples with a magnetic field (blue). Magnetic and electric fields of an electromagnetic wave are perpendicular to one another and to the direction of the wave. While most of the energy carried by these waves is invisible to our eyes, a small portion of the spectrum (wavelengths of 380 to 700 nm). The shorter the wavelength, the higher the energy, therefore every wavelength over the ultraviolet spectrum (including x-rays and gamma rays) are dangerous as they represent ionizing radiation (can remove electrons from the atoms and molecules of materials – for example from air, water and living tissue).

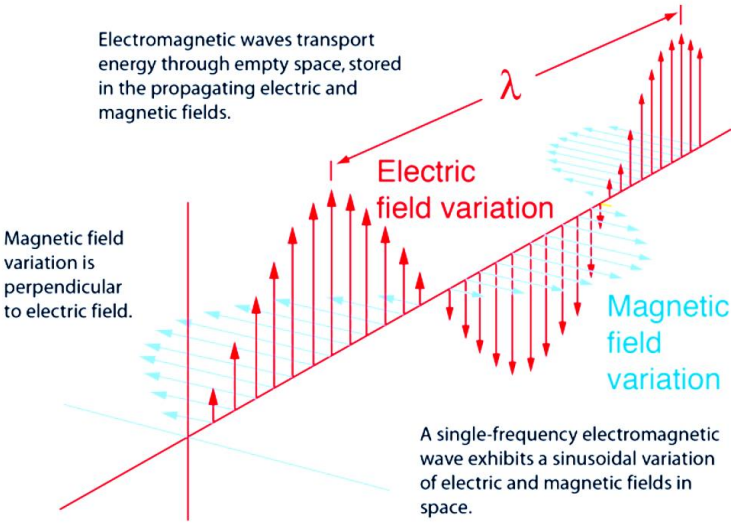


Figure 67. The electromagnetic wave

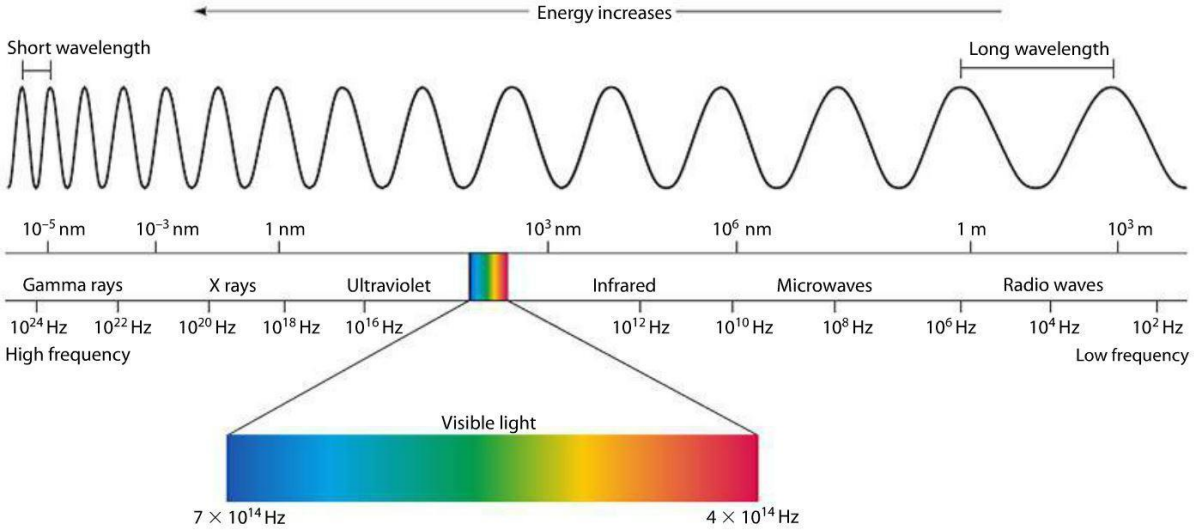


Figure 68. The electromagnetic spectrum.

2.5 Typical analog applications

2.5.1 Filters and filtering

Filtering is the process of eliminating a part of frequency spectrum. Elimination means that specific frequencies pass through the filter and get their amplitude decreased dramatically. The low-pass filter for example shrinks the amplitude of the high-frequency signals, whereas the high-pass does the same for the low-frequency signals. The band-pass eliminates signals that do not belong to a specific band, whereas the band-stop leaves all signals lower than or higher than a specified band, to pass.

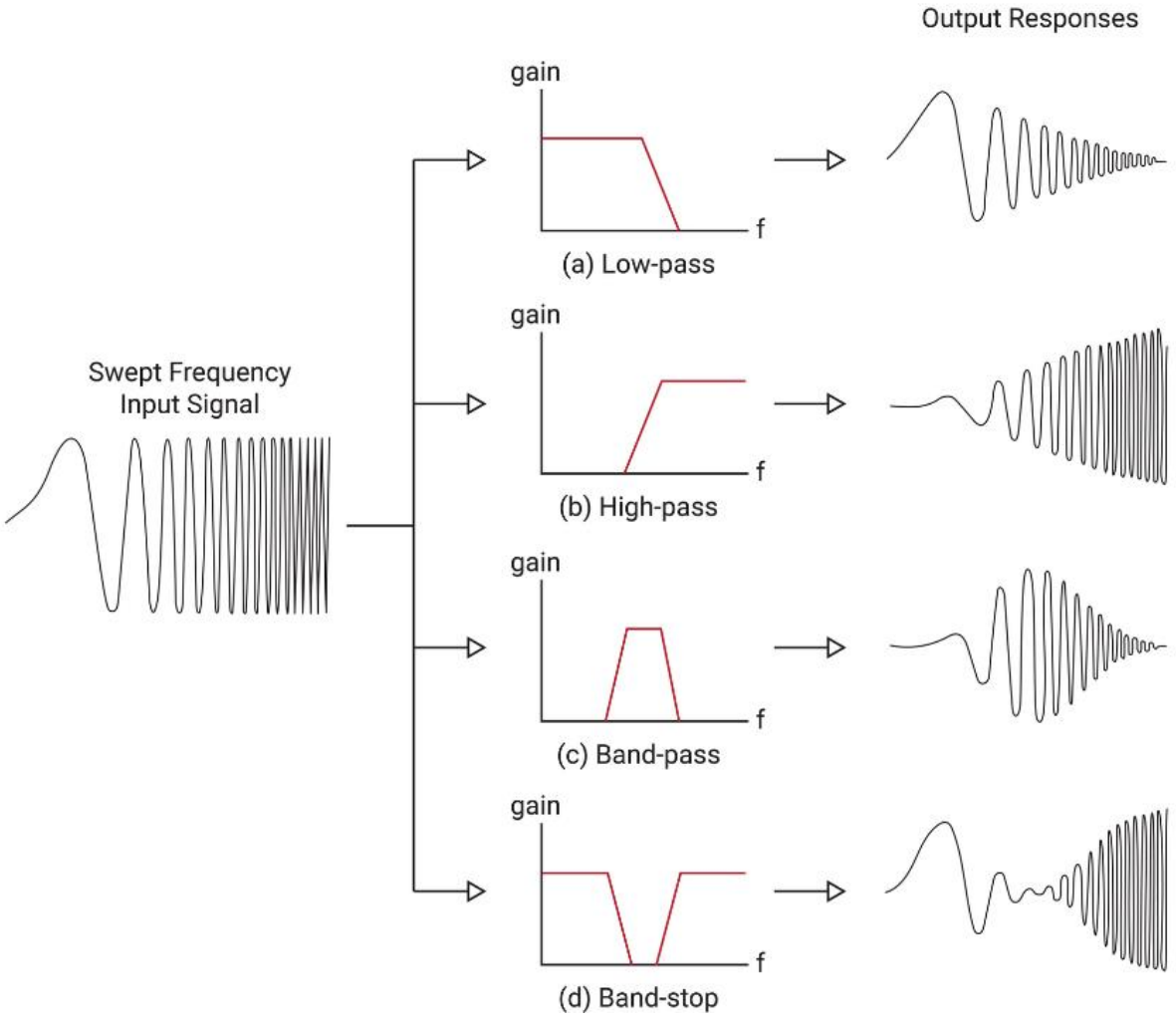


Figure 69. Filter types and their responses

In the above figure, the input signal starts with low frequency content, then it speeds-up. On the right side, one can see the effect of different filters like low-pass, high-pass, band-pass or band-stop. Take the time to think it over, by looking at the output responses.

Often, the filter introduces a phase shift (output will lag or lead the input signal). The attenuation or gain (amplitude change depending on the frequency) and phase shift are easily shown in Bode plots.

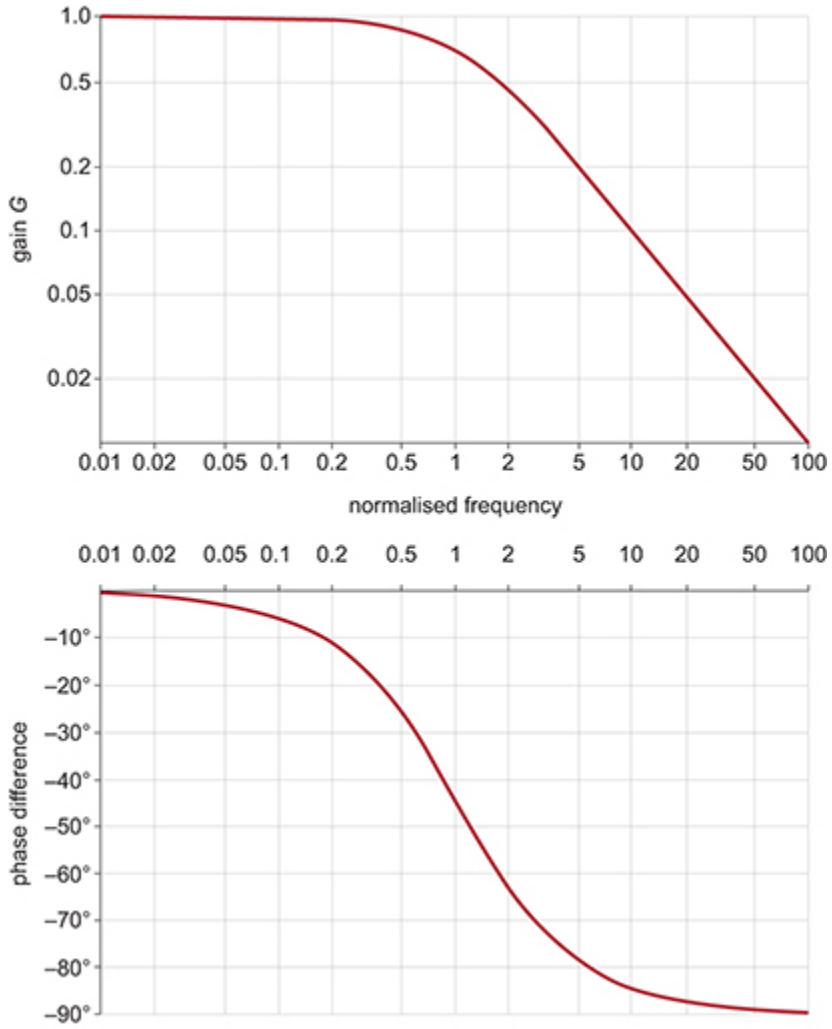


Figure 70. First-order low-pass filter frequency response with gain and phase (Bode plots)

Normalized frequency means 1 is the cut-off frequency.

2.5.2 AC mains. Frequency. Maximum voltage, effective (rms) voltage.

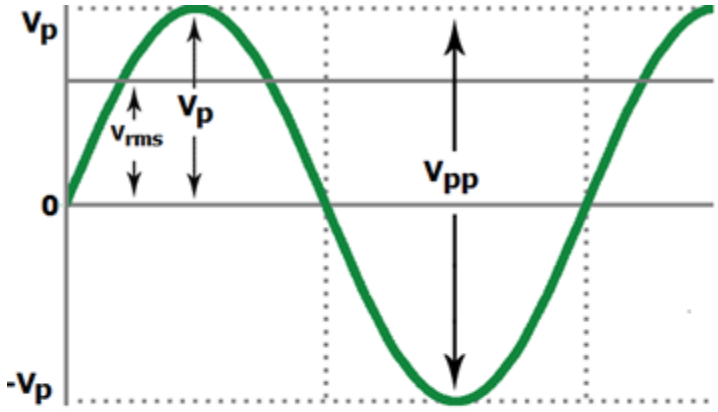
The RMS voltage of an AC voltage is the DC voltage that would produce the same amount of heat. The formula for the RMS value of X is:

$$X_{\text{rms}} = \sqrt{\frac{1}{T}\int_0^T x^2\,dt}$$

Figure 71. The RMS value of X

Assuming X is an AC voltage, therefore X(t) = Vpeak * sin(wt), where Vpeak is the maximum value, w is the angular frequency, and t is time. In Europe, Vrms is most commonly, 230 Volts and the peak voltage (Vpeak) is 325 Volts. Vpp is the peak-to-peak voltage (2*Vp)

Solving the RMS value for the AC sinewave, we get the Vrms = Vpeak / √2



$$V_{rms} = \frac{1}{\sqrt{2}} * V_p = 0.7071 * V_p$$

$$V_{rms} = \frac{1}{2\sqrt{2}} * V_{pp} = 0.35355 * V_{pp}$$

Figure 72. The Vrms, Vpeak and Vpeak-to-peak values of a sine AC voltage

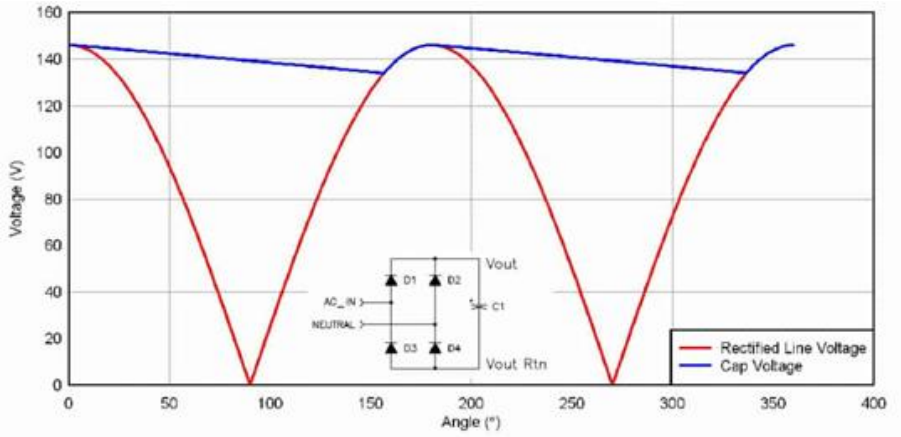


Figure 73. A full wave AC rectifier

In Figure 73

with red, the shape of AC voltage after being rectified by a diode bridge. With blue, the voltage of the capacitor that tries to hold the output voltage constant. In the US, the period of the AC oscilation is 16.6ms (60Hz), while in Europe, it is 20ms (50Hz).

2.5.3 Cubic power supplies

This circuit is very common in older power supplies for mobile phones, network equipment, lamps, pretty much everything with lower than 100W consumption. They are still used in very cheap battery chargers and Christmas tree-lights (for the low price) and high-performance laboratory power supplies (for the very low output noise).

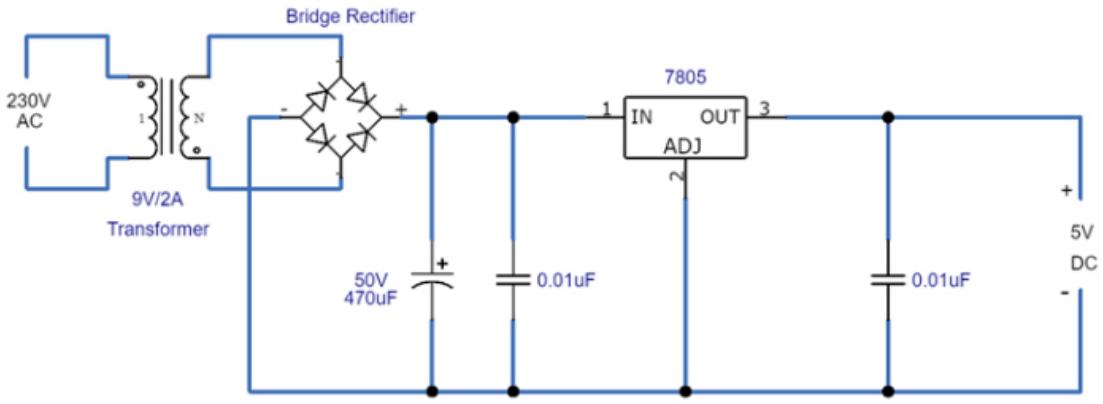


Figure 74. 230VAC-to-5V, transformer-based power supply based on a 7805 IC.

Nowadays, switching-mode power supplies are mostly used in all consumer electronics, where the AC is rectified to DC, then stepped-down using a periodic movement of energy

from input to output, using an inductor or capacitor. The feedback is usually provided by optocoupler.

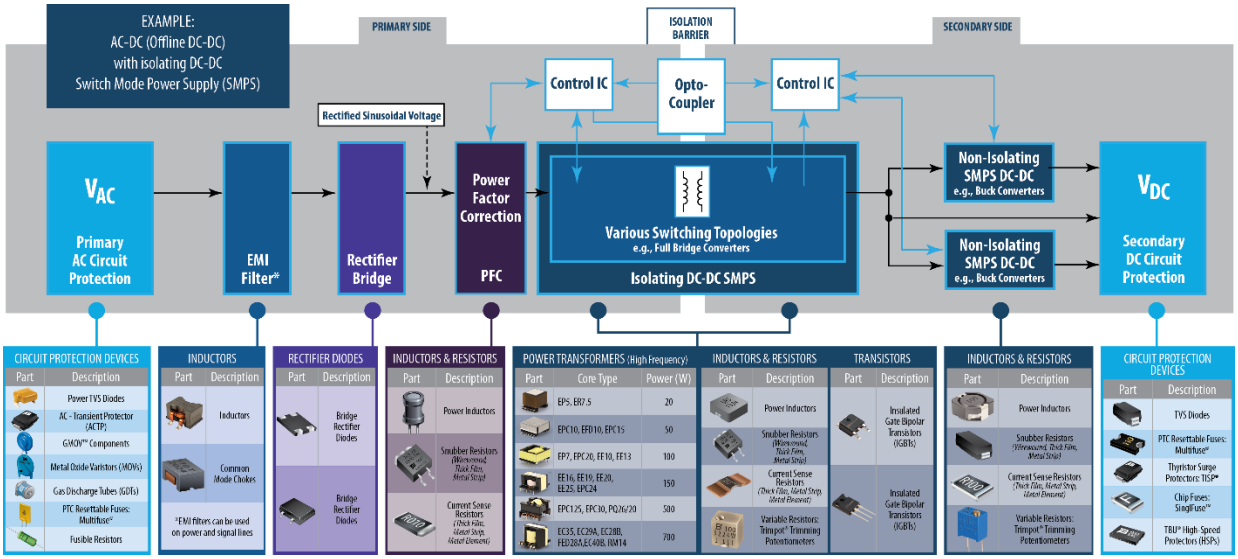


Figure 75. Schematic of a modern PSU with possible component selection

The transformer keeps the same sinewave shape for 230VAC input, but depending on the ratio of wire loops at the primary winding and secondary winding. It is a comfortable way of reducing the voltage to a save level.

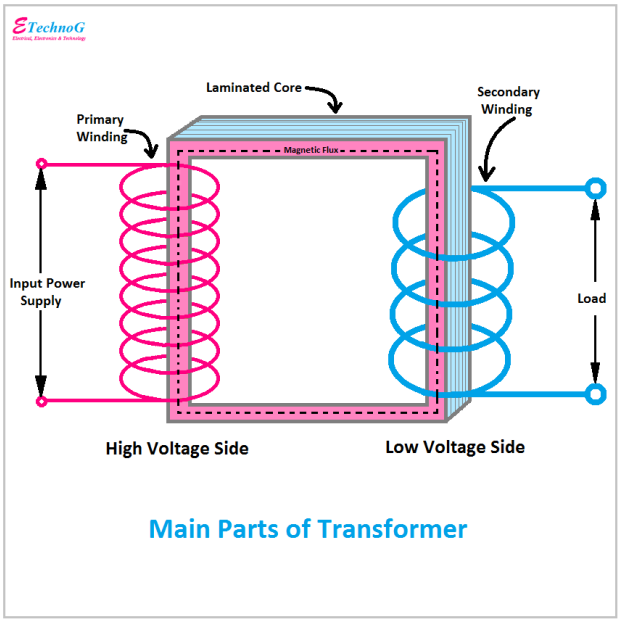


Figure 76. Step-down AC transformer

2.5.4 Step-up and step-down DC/DC converters

There are two types of converters: linear (where the excess power is wasted as heat) and switching (which are usually more than 90% efficient)

Step-up converters: Vout is larger than Vin (very useful for powering electronics from a single battery, eg. mouse from an 1.5V or 1.25V AA battery)

Step-down converters: Vout is smaller that Vin (very useful in carrying large power into a device as high voltage/low-current, then changing it in lower voltage in order to be used by todays' electronics. This is how the phone charger are able to carry large powers (up to 100 W), yet keep the charging cables small: even if the li-ion battery has nominal voltage of 3.7V, the input voltage into the phone may be up to 20V.

The linear step-down is implemented as a series-regulator which wastes excess voltage.

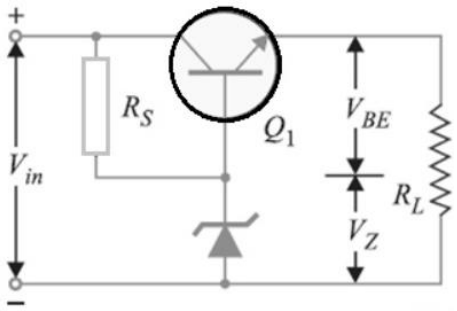


Figure 77. Series-voltage regulator based on a transistor and Zener diode

The above circuit keeps the output (the voltage across the R_L) at a constant $V_{BE} + V_Z$. The difference between the Vin and the Vout is spread across the collector-emitter junction of the Q1 transistor. The Rs is a resistor that ensures the Zener diode stays in a region of breakdown. Assuming the Vin is 9V and the Vout is 5V, this regulator has an efficiency of about 5 / 9 = 55%, which means the battery will last about half the time it would last with a 99% efficiency dc/dc converter. Linear converters are useful even in more expensive equipment, because they are lower-noise than switching regulators.

There are various types of inductor-based DC/DC converters and a few low-power capacitor-based DC/DC converters.

Capacitor-based (inductorless) DC/DC converters are usually used as step-ups, in compact low-power designs. A common type is the charge-pump (which can be extended from voltage doubler to larger voltage multiplications). The voltage doubler works by charging the capacitor to the Vin, then putting the capacitor in series to the output capacitor, increasing the amount of voltage delivered to the load, incrementally; the voltage is delivered toward

the output via a diode, to prevent energy flow from the output back to the input; maximum achievable voltage with this technique, using one capacitor is 2*Vin - Vdiode. It is often used in ICs that have to convert voltages to double/half, like the MAX232 which converts TTL to RS232 levels.

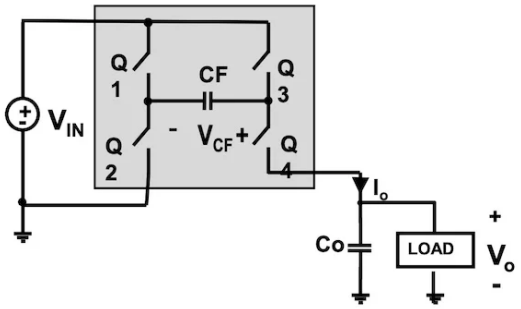


Figure 78. The voltage doubler

Inductor-based DC/DC are used for both stepping-up and stepping down-voltages.

There are multiple topologies, but in non-insulated designs, buck topology is used for step-down and boost topology is used for step-up. A combination of the two exists, the buckboost, for both step-up and step-down.

The Buck converter

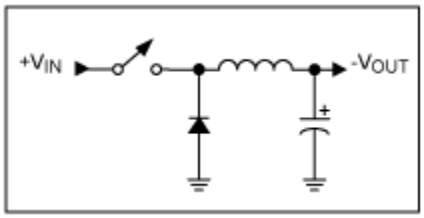


Figure 79. The buck converter topology

The converter uses an inductor to accumulate energy as magnetic field, when the switch is ON: since the inductor will oppose the current flow from input to the output capacitor, it will start accumulate energy. When the switch is OFF, the inductor will oppose the drop of voltage, and reverse polarity of the current, which will now flow though the diode and charge the output capacitor. Depending on the duty cycle (how much time the switch is ON vs ON + OFF), the voltage on the output capacitor may be kept constant (if the load consumes more energy, a higher duty cycle will be needed to replace it, if not, a lower duty cycle will be needed)

The boost converter

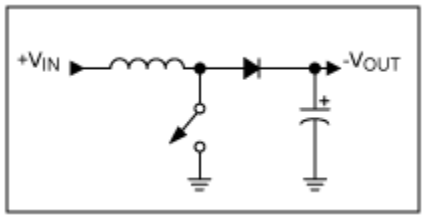


Figure 80. The boost converter topology

The boost converter works similarly to the charge pump. If the switch is set to OFF for a long time, the Vout will be near Vin (minus a small drop onto the diode); The charging phase start when the inductor is grounded (the switch is ON), therefore it will start accumulating energy in its magnetic field. When the switch is set to OFF, the inductor will reverse polarity to oppose the current drop, and will pump energy over the diode into the output capacitor which already was at Vin potential, therefore increasing it over the Vin. This is called the discharge phase. If the inductor is small, we will need a higher on/off frequency on the switch to be able to carry the same amount of energy from input to output (as opposed to larger inductor, smaller frequency)

The buckboost converter (Step-up and step-down)

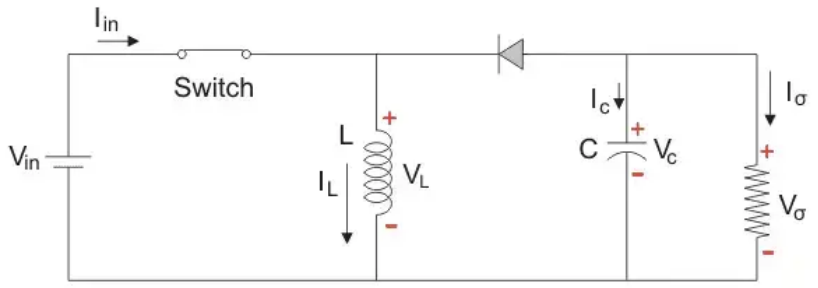


Figure 81. Buckboost in charging stage

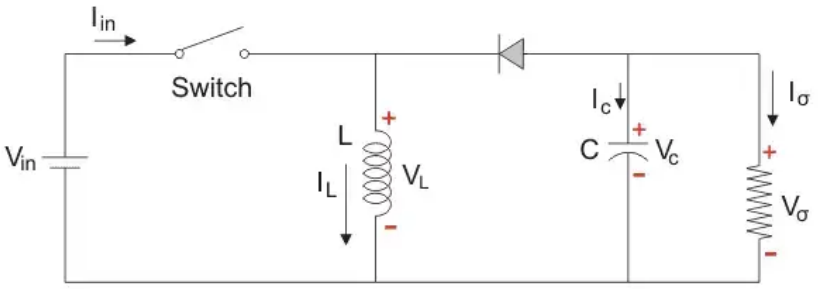


Figure 82. Buckboost in discharging stage

Depending on the duty cycle, more or less amount of energy is delivered from the inductor to the output capacitor, such that the output voltage, may go higher or lower than Vin.

A very versatile IC circuit for Boost / Buck / Inverting is the MC34063. A typical application for it will use very few external components:

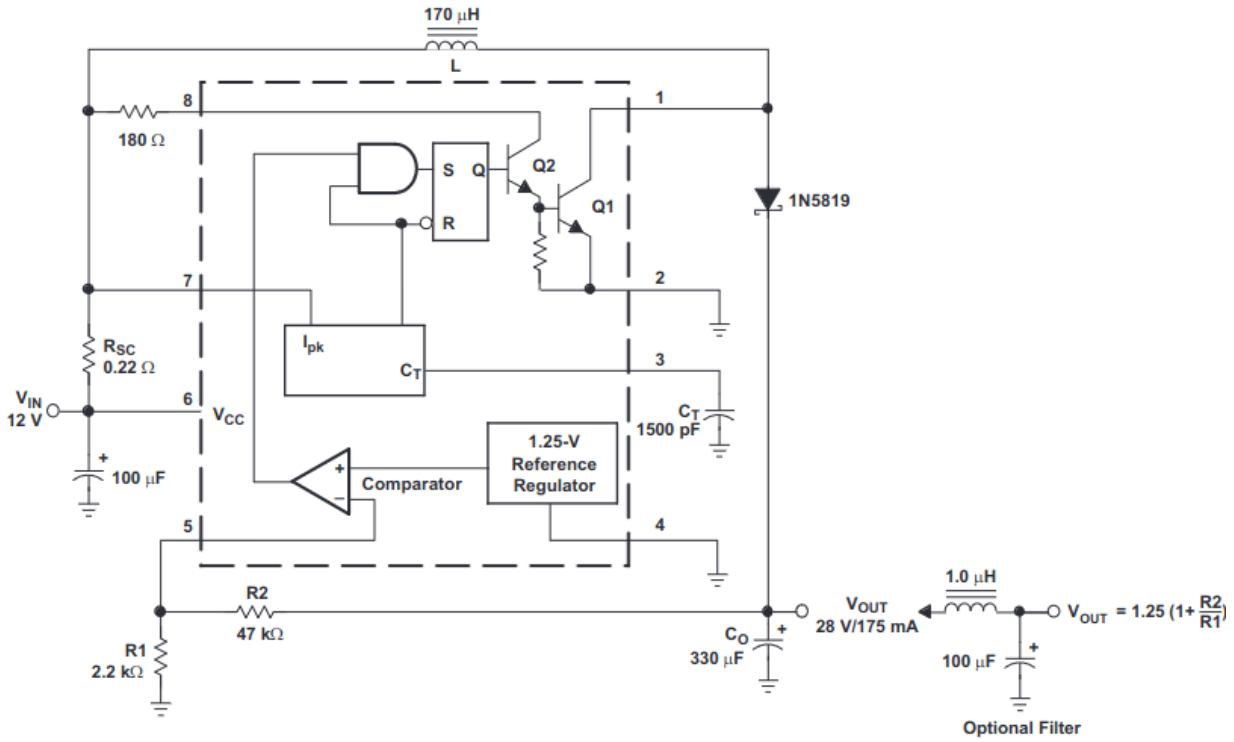


Figure 83. Typical MC34063 usage as step-up converter. Additional output filters may be added to smooth the output

2.6 Batteries

Primary batteries are non-rechargeable batteries: once depleted, they cannot be recharged. Most common primary batteries are:

- Alkaline batteries (more expensive, usually the choice for digital cameras or toys)
- Silver-oxide batteries (watch, thermometers, meters)
- Zinc-air batteries (button format, for hearing aids)
- Lithium coin (e.g. CR2032, for RFID, memory backup, calculators, smart home)
- Zinc-carbon batteries (cheapest, usually in R3, R6, R20 format used for clocks, remote controls)

Applications	Alkaline	Carbon Zinc	Lithium
Access Control System	●		●
Asset Tracking System	●		●
Cameras	●		●
Clocks		●	●
Electronic Door Locks	●		●
Electronic Shavers	●		●
Flashlights	●	●	●
Home Security System	●		●
IoT (Internet of Things)	●		●
Medical Equipment			●
Meters	●	●	●
Personal Care	●	●	
Remote Control		●	
Security Sensors	●	●	●
Smoke Detectors		●	●
Toys	●	●	

Figure 84. GP's selection guide for cylindrical primary battery types

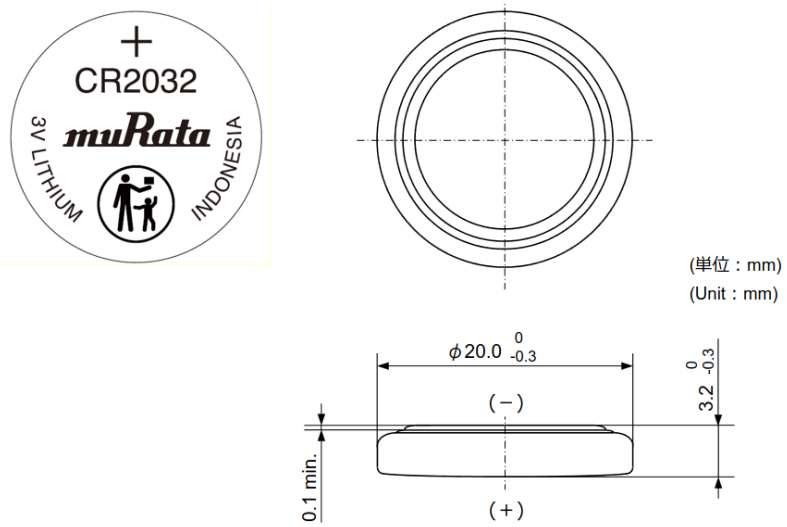


Figure 85. CR2032 coin cell BIOS-backing battery, having 20 mm diameter and 3.2 mm thickness

Secondary batteries are rechargeable batteries, and depending on the technology they can be:

- Li-Ion based (used in laptops and portable tools, energy storage, electric cars)
- Li-Polymer batteries (used in toys, and portable tools)
- NIMH batteries (used in digital cameras, flashes, hybrid cars)
- Lead Acid (used in cars for 12V system, or solar energy storage)

The following problems occur in batteries, and one should balance pros and cons of each technology:

- Self discharge: depending on the temperature, the unused battery will discharge itself over time
- Limited number of charge-discharge cycles
- Maximum depth of discharge
- Behavior at high temperatures (danger of explosion)
- Behavior at low temperatures (temporary loss of capacity)
- Behavior on overcharge (danger of explosion or permanent loss of capacity)

Depending on the chemistry, a different **nominal open-circuit** voltage per element is characteristic to each battery type:

- NiMH have 1.25V / element, which may put into difficulty circuits designed to be powered from typical 1.5V cylindrical cells.
- Li-Ion have 3.6 V, but during charge may get up to 4.2V / element
- Li-Poly have 3.7V / element
- Lead Acid have 2V / element

The energy storage capacity is expressed in Wh (power delivered in a certain amount of time). The (charge) storage capacity is also expressed in Ah (current delivered in a certain amount of time).

The maximum charge or discharge current is expressed in Amperes, or C-units (a C-unit is the current needed to fully charge or discharge a battery in one hour)

An example of such a battery would be a typical lead-acid car battery, having 55Ah @ 12V (6 elements of 2V connected in series). These batteries do not like deep-discharges, but the are content with float charging (keeping them at a certain voltage indefinite time). Optimized AGM batteries may be discharged 50%, and have a cycle life of 800.

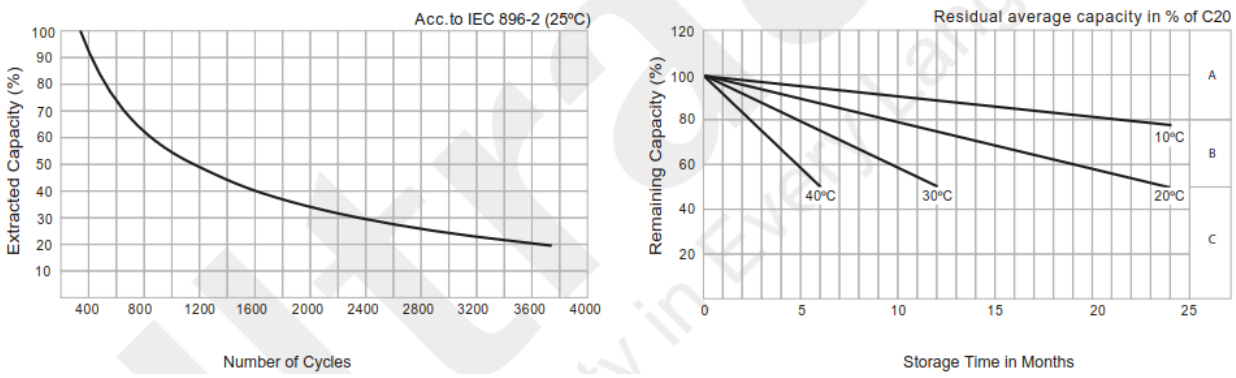


Figure 86. Cycle life vs depth of discharge (left). Storage time vs temperature

Optimized solar storage batteries, like UCG100-12 (100 Ah, 12V) may reach up to 3600 cycles, assuming extracted capacity does not go over 20%.

The discharge current will also influence the amount of energy one can extract from the battery. For example, in the below figure, discharging at 0.1C (10A for the UCG100-12) one can extract energy for 10 hours => 10A * 10h = 100 Ah, 100% nominal capacity, whereas extracting energy at 2C (200A for UCG100-12) will only last 10 minutes => 200A * 1/6 hours = 33.3 Ah, a mere 33% of nominal capacity.

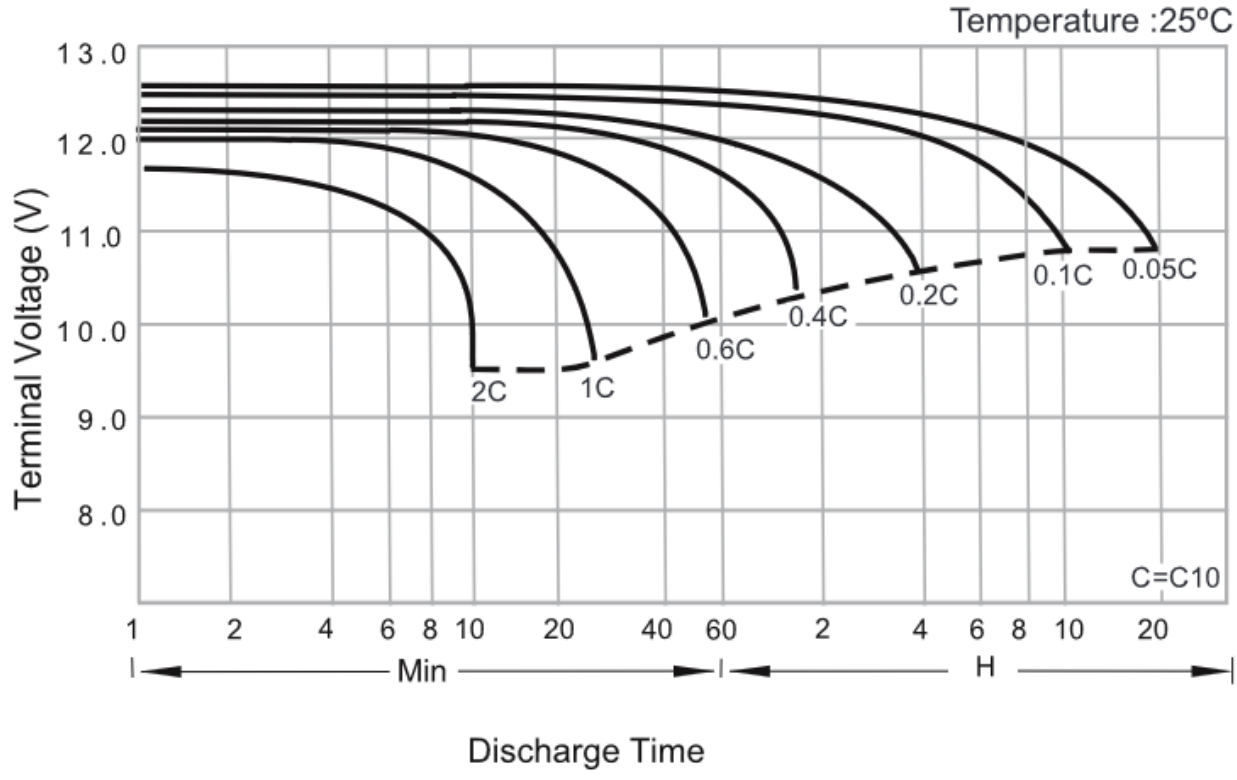


Figure 87. Effect of discharge current, on capacity in UCG100-12 lead-acid battery

The advantages of lead-acid

In higher performance, battery packs, various combinations of voltages and maximum currents are needed. Therefore, to increase the voltage of a battery pack, multiple battery elements may be connected in series. To increase the maximum current delivered by the battery pack, multiple battery elements may be connected in parallel. To express a connection scheme, one uses the terms xxSyyP which means xx element in series of yy in parallel: total number of elements: xx*yy. For example, a 4S3P is 4 blocks of 3-parallel cells.

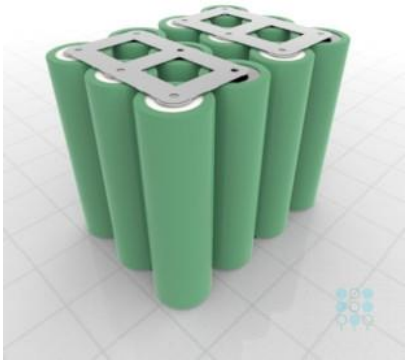


Figure 88. 4S3P pack. Note the white top part and the grey top part.

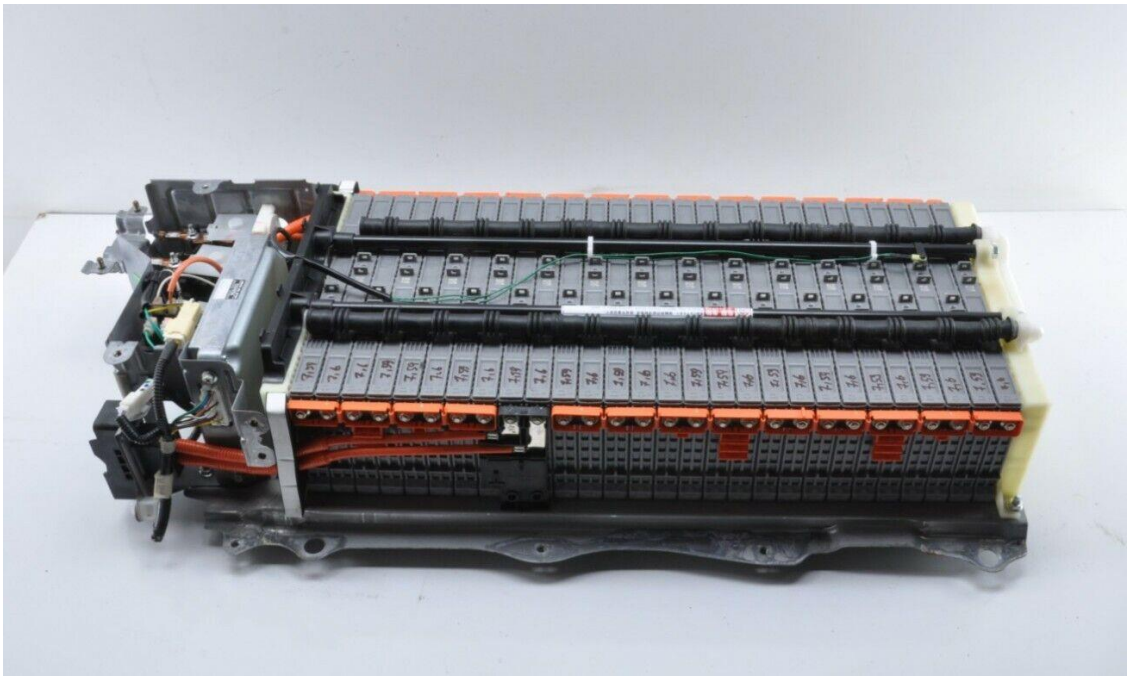


*Figure 89. Toyota Prius 2 Hybrid battery: 28 modules of 6 cells each having 1.2V and 6.5Ah NiMH. Total voltage: 28*6*1.2 = 201.6 V nominal. Total pack energy: 201.6 V * 6.5 Ah = 1.3 kWh.*

Keeping all batteries at similar parameters (charge), requires a process names balancing. If no balancing is performed, in time, one of the series-mounted cells may discharge more than the others, keeping the pack at poor performance. Active balancing recovers energy from the top-performing batteries and redistributes it to the others, while passive balancing, wastes it as heat.

Figure 90. Snapshot of Dr.Prius app, which reads voltages across all 14 battery packs of Prius 3, in order to diagnose "bad" cells.

Figure 91. Very common laptop-battery element (li-Ion 18650 format)

The Li-Ion 18650 name refers to diameter (18 mm) and length 65.0 mm. The -35E refers to capacity, 3500 mAh.

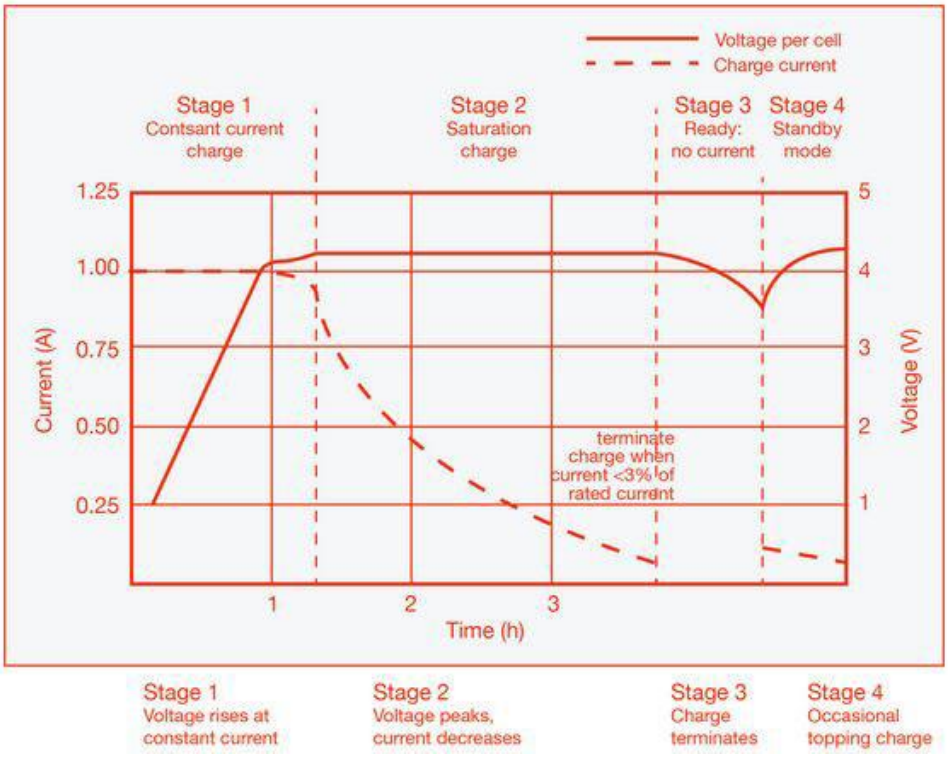


Figure 92. A typical algorithm for charging Li-Ion batteries. Constant current, then constant voltage.

The reason for the manufacturer's specification of 10%-80% charging time is given by the maximum energy transfer during this time. After 80% charge, the rate of charge must be decreased to protect the battery and keep the temperature low.

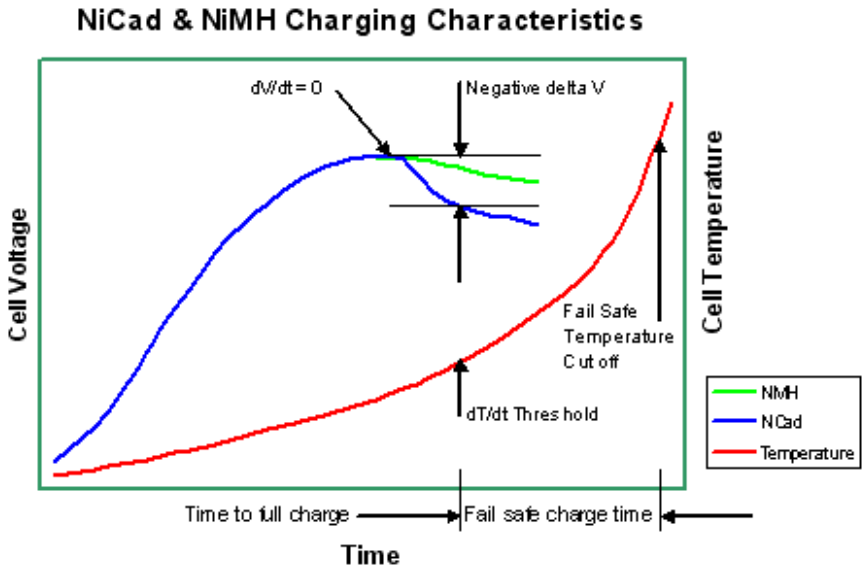


Figure 93. NiCd and NiMH dv/dt and dT/dt marking the end of charge

The negative delta V is a common way of detecting end-of-charge in NiMH batteries. A safety feature is the high temperature cutoff.

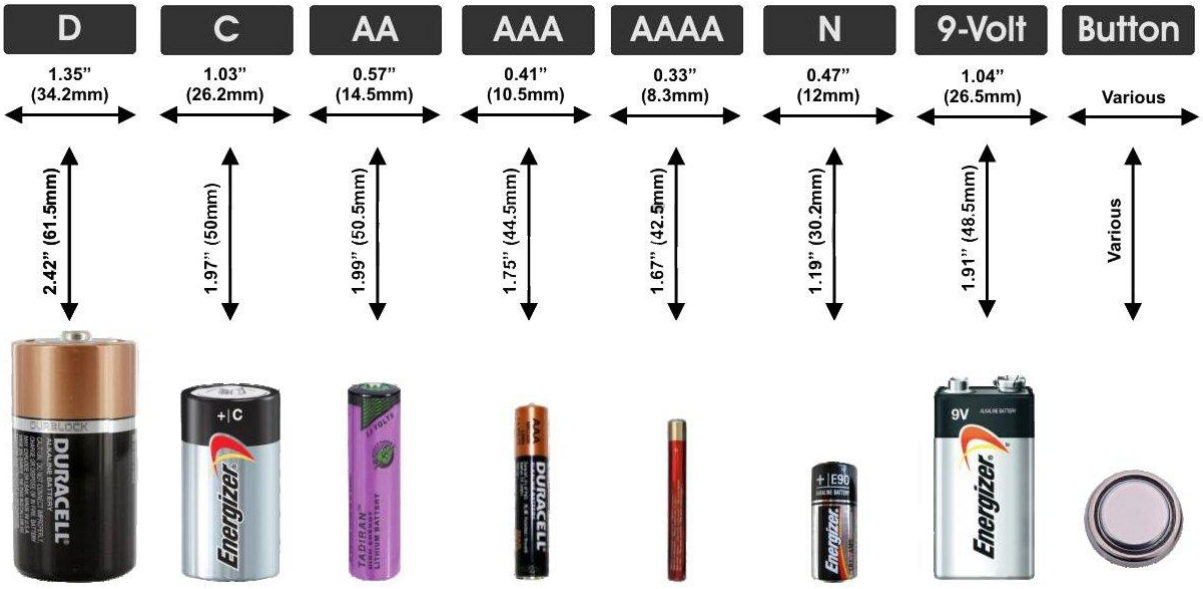


Figure 94. Common physical formats for batteries.

Energy density and power densities are summarized below:

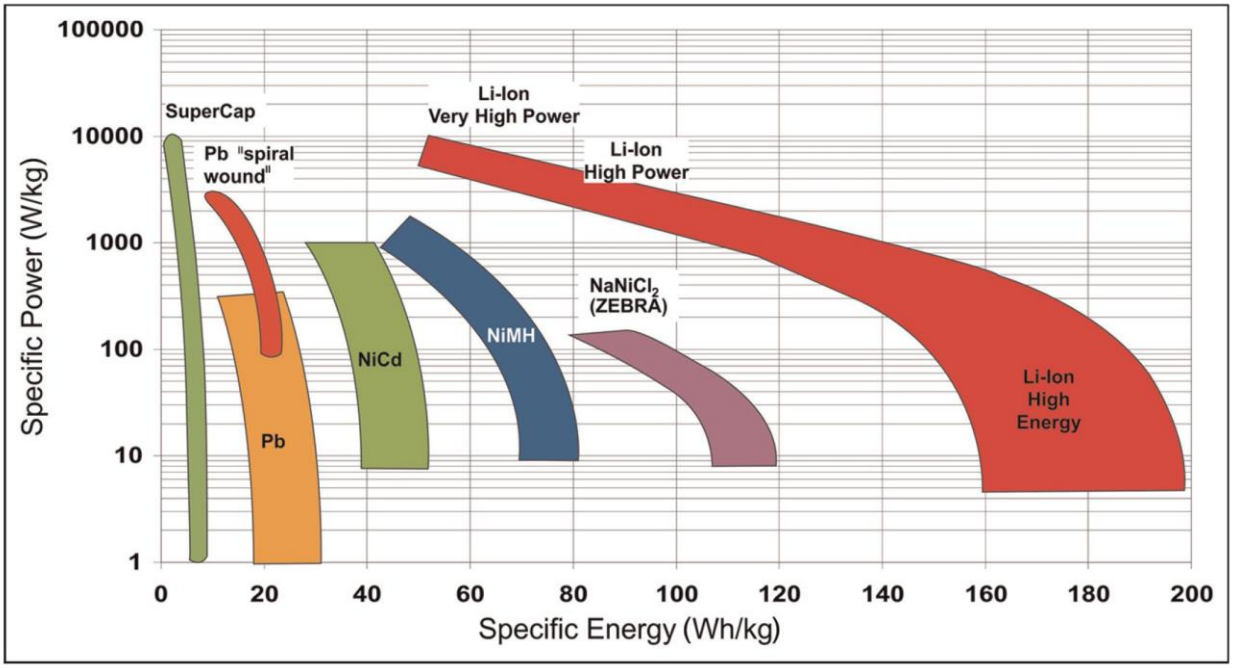


Figure 95. Summary for specific energy and specific power across all common battery chemistries

2.7 Data transmission and conversion

Depending on the time data is transferred from sender to receiver and back, the communication interfaces are of three types:

- Simple (one direction only)
- Full-duplex (both directions, simultaneously)
- Half-duplex (both directions, alternatively)

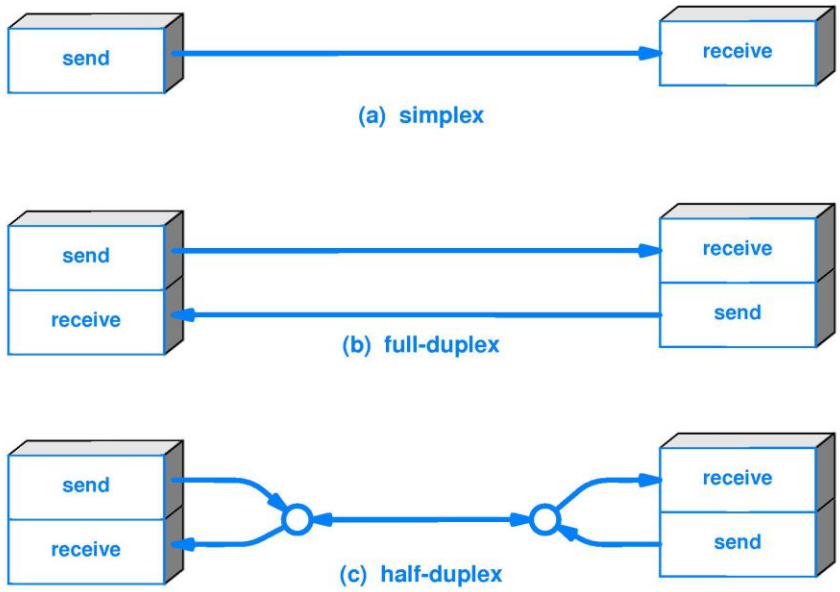


Figure 96. Data transmission types

If a synchronization mechanism exists based on an explicit signal sent from one partner to the other, communication protocols may be:

- Synchronous (USART, SPI...)
- Asynchronous (UART, I2C, USB...)

Depending on the number of bits transferred at a time, the communication protocols may be:

- Serial protocols (low number of wires, higher data rate implies GHz frequencies)
- Parallel protocols (higher number of wires, originally meant for higher speed)

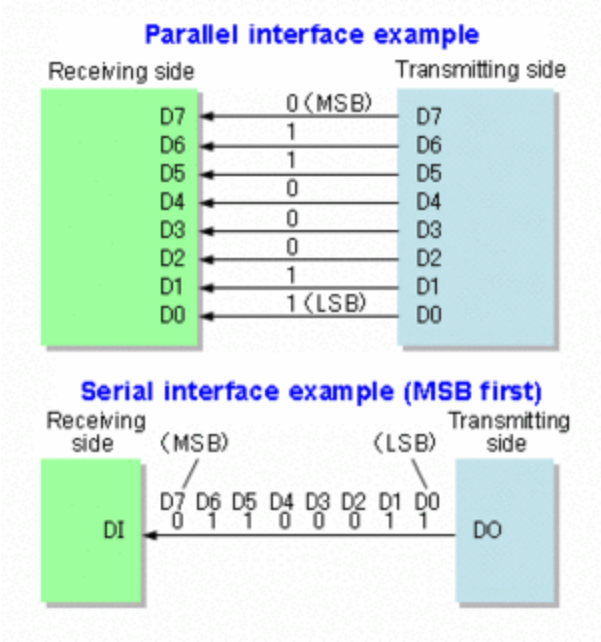


Figure 97. Parallel vs Serial interfaces

2.7.1 SPI communication protocol

There are multiple flavors of the SPI communication protocol: usually there are 3 wires for it:

- clock (SCK, output by master)
- master out slave in (MOSI, data line output by master)
- master in slave out (MISO, data line output by slaves)
- an optional slave select line (SS) for each slave sharing the bus

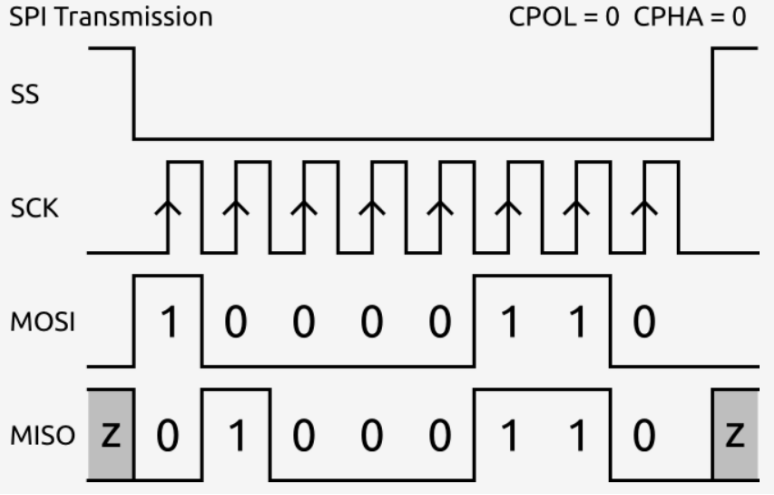


Figure 98. Example of a SPI communication (8-bit per frame)

As one can see, the SPI is a synchronous (has clock), serial (1-bit data) protocol, full-duplex (MOSI and MISO can have different values at any time)

Depending on the edge the data is output by the master and sampled by the slave, there are 4 types of SPI modes (CPOL = clock polarity, CPHA = clock phase) :

M0: CPOL=0, CPHA=0 (clock is default low, and MOSI is sampled on the rising edge of it)
M1: CPOL=0, CPHA=1(clock is default low, and MOSI is sampled on the falling edge of it)
M2: CPOL=1, CPHA=0 (clock is default high, and MOSI is sampled on the rising edge of it)
M3: CPOL=1, CPHA=1 (clock is default high, and MOSI is sampled on the falling edge of it)

Common speeds are 12, 25 and 50 MHz (and since the MOSI line can send one bit per clock cycle, it would reach i.e. 12 / 25 / 50 Mbit/s). If there are multiple slaves listening to the same SCK and MOSI lines, they will be selected by the master, using a dedicated SS line (for each slave), and only the selected slave will answer with the MISO line change.
The SPI bus is used in interfacing the SD cards in the slow compatible mode of 25 Mbps.

There are two variations of the SPI:

- the Dual SPI where the MOSI and MISO lines become bidirectional such that each device may send 2 bits of data per clock cycle
- the Quad SPI (QSPI) where 4 bidirectional data lines are used to transfer 4 bits of data per clock cycle
- the Octal SPI where 8 bidirectional data line are used

Additionally, a DDR mode (dual data rate) exists, for transferring data on both rising and falling edge of the clock.

2.7.2 I2C communication protocol

The I2C (IIC, or inter-integrated circuit) protocol was originally developed by Philips Semiconductor in 1982. Some chip manufacturers call it TWI (two wire interface).

The two wires are called SDA (data) and SCL (clock), so it is a synchronous bus, half-duplex. It allows multiple masters but is usually used in a single master – multiple slaves scenarios. Slaves distinguish themselves by address which may be 7-bit or 10-bit address.
There is a special condition called “start”, to mark the beginning of a frame and “stop”, to mark the end of a frame.

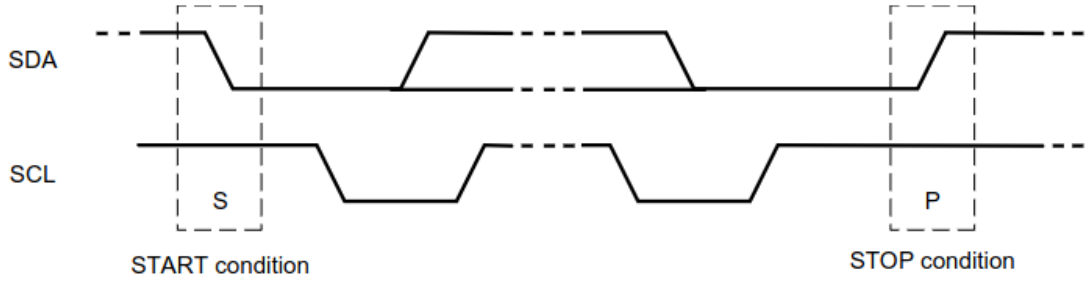


Figure 99. START and STOP conditions on I2C bus

SCL and SDA are open-drain outputs with external pull-up resistors. SCL is clocked by the master and SDA is output by both masters and slaves in half-duplex manner. Since the SCL are open-drain outputs, SCL line may also be kept in low level (clock-stretching) by a slow slave, to mark the fact cannot output data in time, and has to delay the master. A special bit of ACK or NACK is issued by the slave (or bus, via pull-up resistors) when the slave acknowledges various conditions.

A typical 7-bit write frame looks like: START, 7-bit address with one bit for W (0), one bit of ACK from the present slave (or NACK if it is missing), then the data from master to slave, with ACK after each byte, and then a STOP.

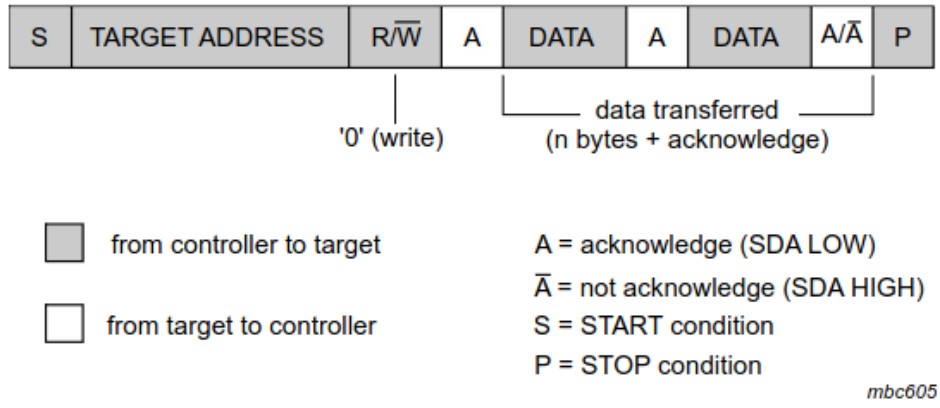


Figure 100 Master sends data to slave (write frame) with 7-bit address

A typical 7-bit read frame contains: START, 7-bit address with one bit for W(0), one bit of ACK from the present slave (or NACK if it is missing), then data from slave to master with ACK from slave, then a STOP.

A typical 7-bit combined read and write frame looks like: START, 7-bit address with one bit for W(0), data bytes from master to slave, one bit of START (called repeated start), 7-bit-address with one bit of R(1), data bytes from slave to master, then NACK, then STOP. All data bytes are ACK except the last one which ends in NACK.

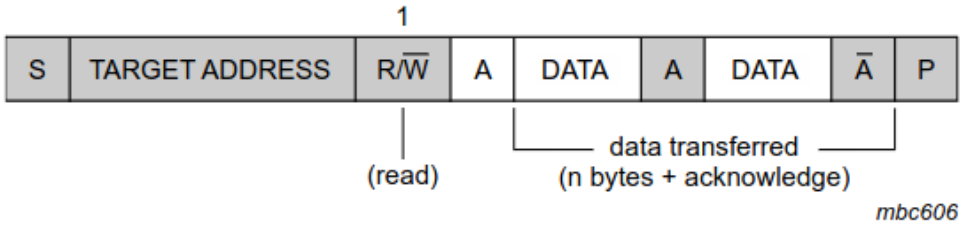


Figure 101. Master reads data from slave, with 7-bit address (read frame)

Table 17. I2C protocol speeds

I2C protocol name	Protocol speed (clock frequency)	Recommended pull-up resistor value (Ohm)**	Maximum risetime (ns)
Standard mode	up to 100 Kbps (kHz)	4k7	1000
Fast mode	100 – 400 Kbps (kHz)	2k2	300
Fast mode plus	400 – 1000 Kbps (kHz)	1k	120
High speed mode*	1000 – 3400 Kbps (kHz)	Active + 1k	80

*High speed mode required active high-drive from master, not just a small-value pull-up resistor.
**The pull-up resistor value depends on the number of devices on the bus (by the capacitance accumulated on the bus)

The I2C protocol is utilized in flash memories, slow ADCs, DACs, microcontroller, LED controllers, GPIO expanders, ASIC controllers...

2.7.3 U(S)ART communication protocol

The U(S)ART communication protocol is one of the oldest communication protocols. The UART is asynchronous (Universal Async Receiver Transmitter), while the USART is synchronous (has a line of clock). Both are 1:1, so no multiple masters or slaves on the same bus, only point-to-point.

Depending on the voltage levels, one might find this protocol as RS232 (with the HIGH/1 as negative voltage and LOW/0 as positive voltage – "inverted UART"). The MAX232 chip is usually used as level translator to change from RS232 to 5Volt UART and back, and MAX3232 for 3V UART; both chips used charge-pumps to boost the voltages as required by RS232 standard.

The U(S)ART may be full-duplex (when RX line is different that TX line) or half-duplex (when RX line is the same as TX line). The half-duplex USART is heavily used in smartcards, while the full-duplex UART is used in microcontrollers and PCs, FIFOs.

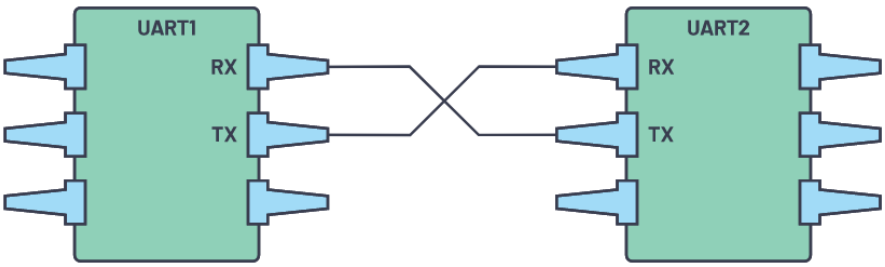


Figure 102. Two partners exchanging data over a full-duplex UART bus. GND is tied together, but not shown.

Common data rates in UART are 9600, 19200, 38400, 115200, 921600, 1M, 3M bits per second. Usually the UART peripheral in a microcontroller implements the data rate as configurable: the peripheral is clocked with 1-100 MHz, then divided to achieve the desired data rate. In smartcards, the default speed is 9600 obtained as 3.57 MHz clock divided by 372.

A data frame is comprised of one START bit, 5-9 data bits, 0 or 1 parity bits and 1, 1.5 or 2 STOP bits. Common configurations are:

- 9600 8E2 (9600 bps, 8 data bits, even parity bit, 2 stop bit)
- 115200 8N1 (115200 bps, 8 data bits, no parity bit, 1 stop bit)

Figure 103. UART data frame

The parity is computed as a 1-bit sum of each bit including the parity bit. If the parity is set to even, that sum must be 0. If it is odd, that sum must be 1.

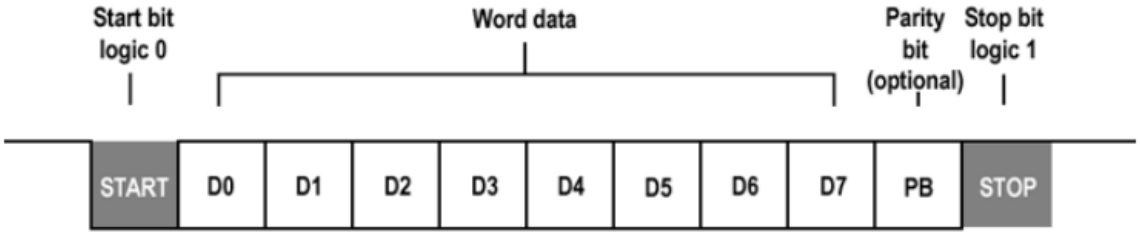


Figure 104. A UART data frame with START bit, 8 data bits, one parity bit and STOP bit.

The bits are sampled usually one, in the middle of each bit. Some microcontrollers will sample the data 3 or 5 times per bit, to make sure the data is read correctly (by majority vote). Some microcontrollers having UART peripheral allow USB communication with a PC by utilizing a specialized chip, like FT232 to convert from USB protocol to UART.

2.8 Microcontrollers and processors

Processors are integrated circuits designed to perform operations on external data source, whereas the microcontrollers (uC or MCU) are integrated circuits design to enforce operations in embedded systems (they contain a processor, memories, and various peripherals).

There are two major architecture types depending on the memory used for programs and data:

- Separate memory buses (and memories) which is called Harvard architecture
- Same memory bus (and memory) which is called von Neumann from the famous computer scientist, engineer, physicist and mathematician, John von Neumann (1903 – 1957). The Von Neumann bottleneck refers to this problem of not being able to transfer data and instructions in the same time (as they are contained in the same memory, and delivered via same bus)

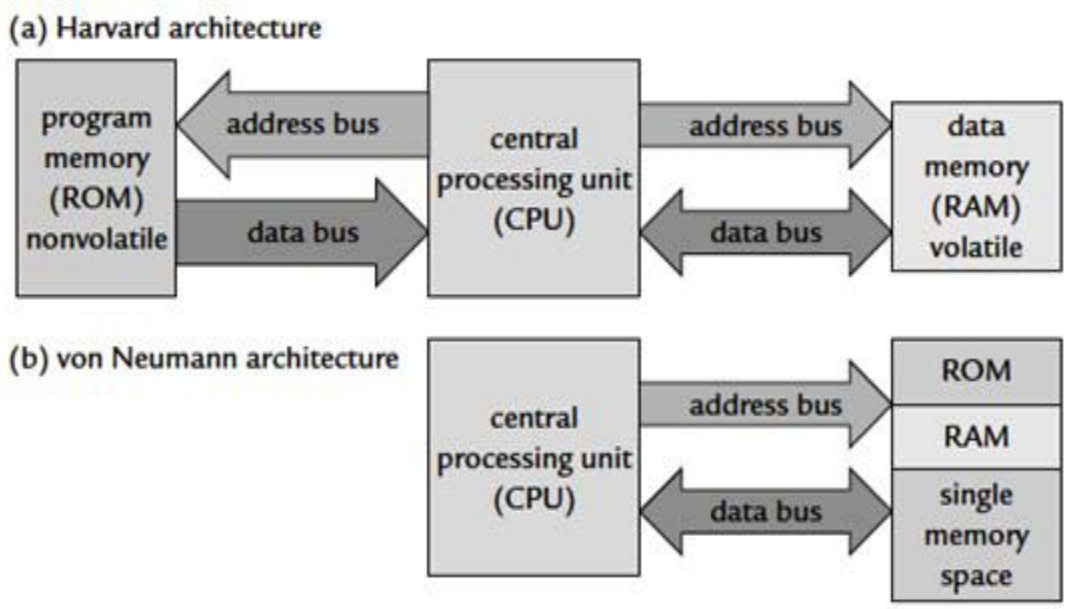


Figure 105. Von Neumann and Harvard Architectures

Companies use computing cores of various architectures (usually Harvard, where the program memory bus and data memory bus are independent as the memories are different), they add flash memories (for programs) and RAM memories ("working memories") for data, and depending on the application, they also add various peripherals for data acquisition or communication. Such an example is the STM32 line from STMicroelectronics, which is based on Cortex M0, M0+, M3, M33, M4 or M7 depending on the market. Microchip offers PIC10F/12F/16F/18F/24F families for simpler tasks and faster 32-bit microcontrollers for

demanding tasks. Texas Instruments (TI) is famous for its 16-bit line of MSP430 ultra-low power MCUs and high-resolution data acquisition engines (analog to digital converters).

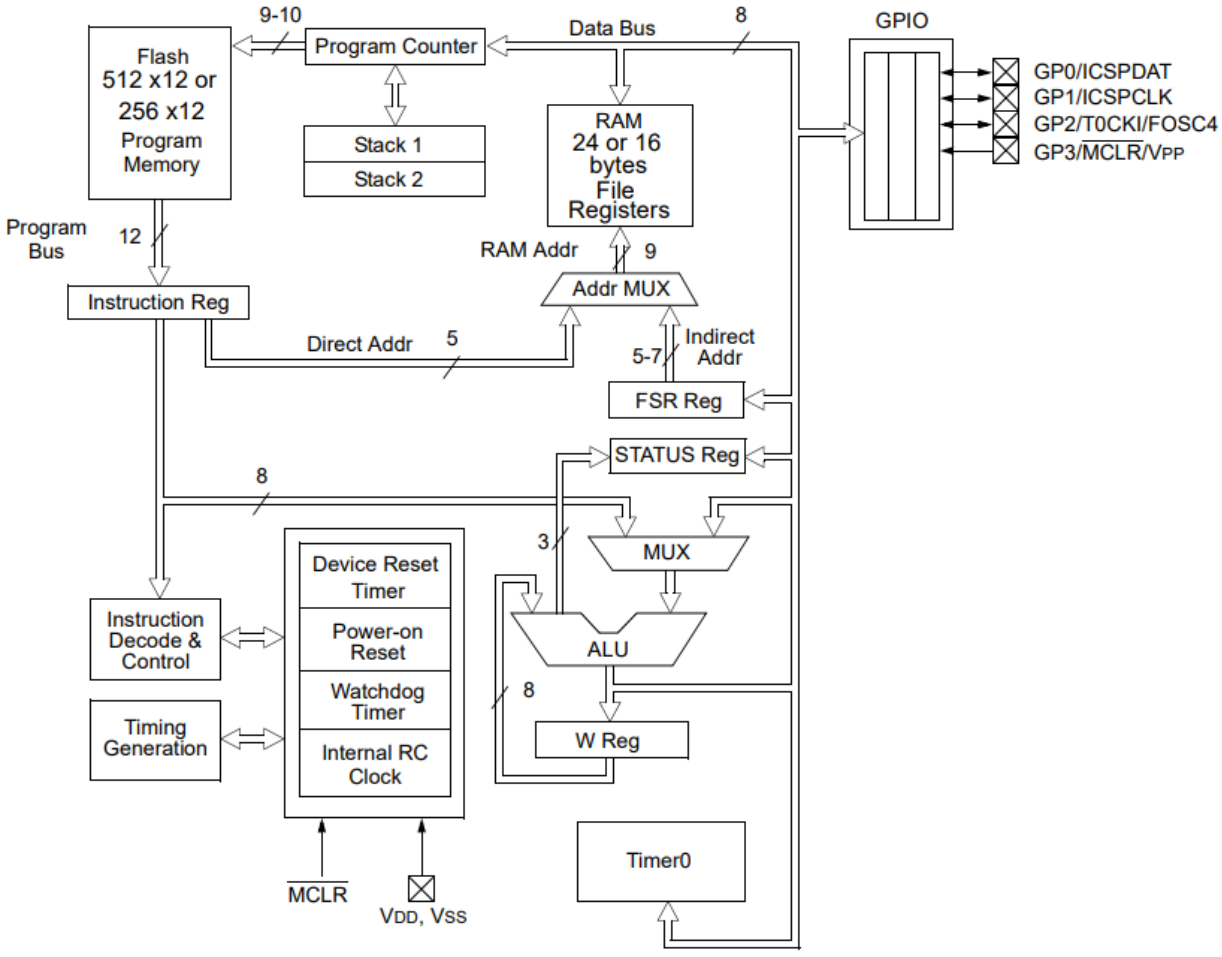


Figure 106. PIC10F2xx internal block diagram

The PIC10F200 is the most low-cost microcontroller from Microchip, and it contains 16x 8-bit registers as RAM storage, 256 program words in program flash memory, an 8-bit timer (Timer0), a WDT (watchdog timer) and 3 I/O pins and one input-only pin, and a 2-level stack. It runs at 4 MHz and can execute most instructions in 1 microseconds (except program branches and jumps). As one can see, it is a Harvard architecture (there is a separate 8-bit data bus, and a separate 12-bit program bus).

The stack is the memory space used either to store return address and sometimes arguments when calling functions. It can be hardware (dedicated separated memory from RAM) or software (has a reserved memory space in the RAM and is accessed using a SP-stack pointer register, which holds a pointer to the last used location or first free location).

Next, is a list of hardware resources a microcontroller may offer:

2.8.1 Resources: I/Os (input/output pins)

The I/Os or input/output pins are pins capable of setting a value (output) of low-voltage or high voltage, or reading an external value (input). They are usually controlled by a dedicated register for the direction (usually called DDR or TRIS) and another one for value (usually called PORT / PIN / GPIO). The high-output value usually is the Vcc where the MCU is powered from, whereas the low-value is GND. Most microcontrollers have protection diodes (clamping diodes) on their input: when the voltage applied goes over Vcc (5V in the example), the top diode opens and increases current consumption towards Vcc to keep the voltage around Vcc + 0.3V. If the voltage goes under the GND potential, with more than 0.3V, the lower diode opens and sinks current into GND.

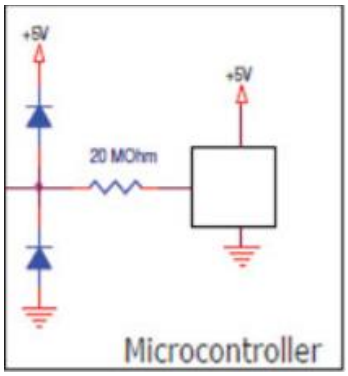


Figure 107. Clamping diodes protect the I/O on the left

2.8.2 Resources: Prescaler

The prescaler is a clock divider which enlarges the input clock period by an integer factor. The slower output clock may be used in other peripherals like timer, WDT or UART or any other slower peripheral.

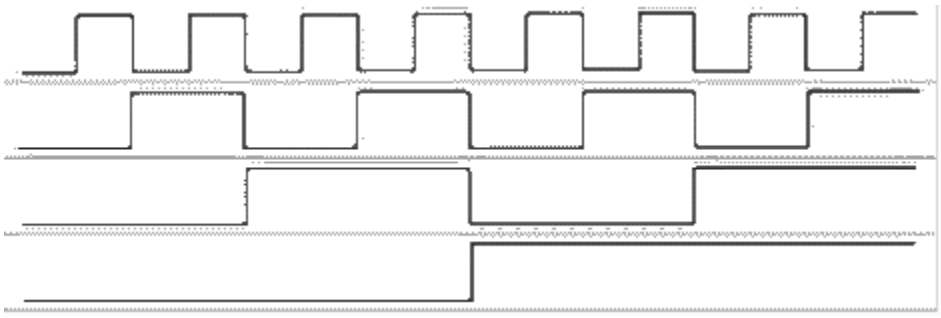


Figure 108. Prescaler input (top), and output for 1:2, 1:4 or 1:8 setting

2.8.3 Resources: Timer

A timer (or counter) is a peripheral that counts the time passed from its start. Since it is a digital circuit, it will count in number of clock cycles. Assume an 8-bit timer (0 – 255 of value) counts starting from 0, upwards, on an 1 MHz clock (1 us clock period): it will overflow after 256 microseconds. When overflowing, the timer either triggers an interrupt, or marks a bit in a status register, or both. Timers are usually used in conjunction with prescalers to be able to count (albeit with lower precision) longer periods of time.

Timers are either simple timers (that count up from 0 to top), preload timers (from a preloaded value to top) and may be coupled with other digital blocks to implement complex behavior on the GPIO pins. Some timers may also count downwards.

Depending on the size of the counting register, a time may count 256 steps (8-bit), 65536 steps (16-bit) or about 4.2 billion steps (32-bit).

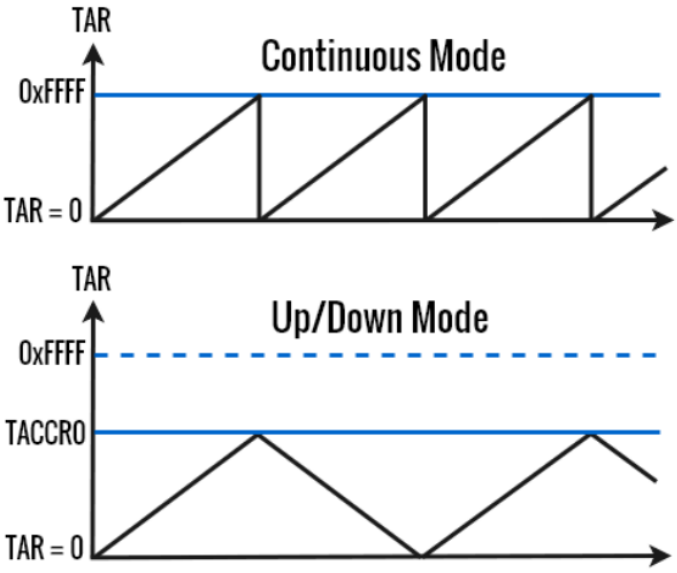


Figure 109. MSP430 MCU timer modes of operation

The MSP430's timer is a 16-bit timer (hence the 0xFFFF), which may count upwards or downwards (from a reload value set in TAR(timer counter register) to a presetable value set in TACCR0 – timer compare register) in bottom figure, or may count upwards from 0 to top (0xFFFF) and overflow to 0 continuously (top figure)

Most microcontrollers have multiple hardware timers, but if that is not available of a specific derivative, one can implement software timers on top of a hardware timer.

2.8.3 Resources: Watchdog timer (WDT)

The WDT is a timer that runs whenever it is activated, and, if not served in time, when it expires it will reset the chip. This is useful to prevent wrong operation (code execution is abnormal, and it stops going through expected program sections).

2.8.4 Resources: Brown-out detect (BOD)

The BOD circuit is a circuit that resets the MCU, if lower (than allowed) Vcc voltage is observed: this is useful as the MCU might not function correctly at lower Vcc than recommended (e.g. Vbot- voltage). The BOD will allow MCU to exit reset state after a certain timeout counted from Vcc recovered to at least Vbot+ voltage.

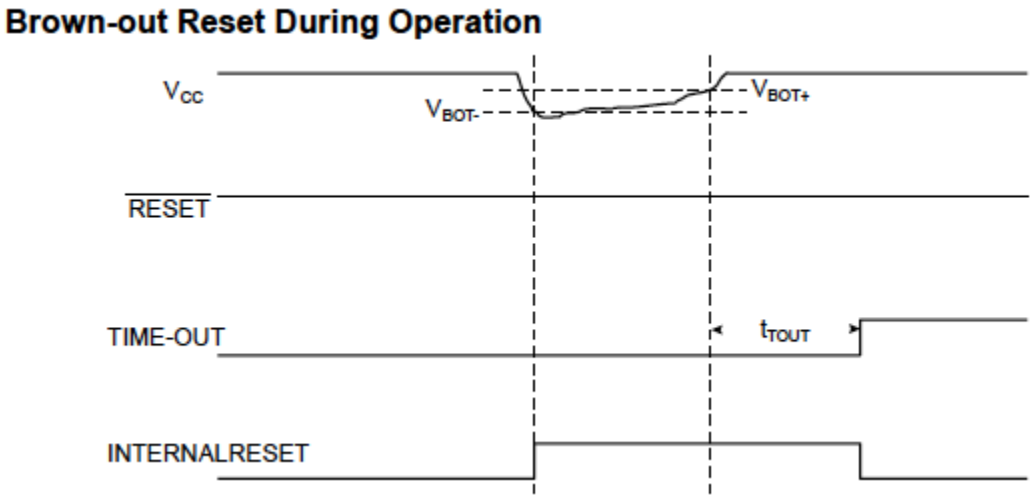


Figure 110. Brown-out detector scenario

2.8.5 Resources: Analog to Digital Converter (ADC)

The ADC is a peripheral that allows to convert a voltage to a number. Depending on the architecture of the ADC some are slower, and some are faster or more precise (but more expensive as they are larger). One of the most common ADCs found in MCUs, is the SAR (successive approximation register) ADC. The N-bit SAR ADC does the conversion in about N steps. A SAR ADC converts the voltage to number as follows:

- Track the input voltage, then switch to hold to keep it stable as Vin
- At each of the N steps: at each step, a bit new is set (starting from MSB to LSB) and the resulting voltage (using a DAC) is compared with the actual Vin: if the DAC voltage is higher, then the bit is reset to 0, otherwise it is kept at 1. Then go to the next bit...

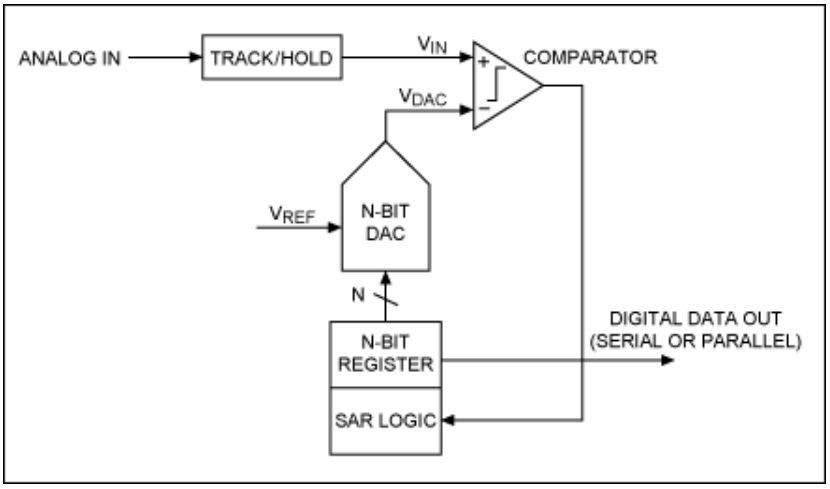


Figure 111 N-bit SAR ADC architecture

2.8.6 Resources: Digital to Analog Converter (DAC)

The DAC is a peripheral that allows to convert a number into a voltage. There are multiple architectures of DACs but one of the most common in MCUs is the R2R DAC. The advantage of such circuit is that it is very simple to build: it needs only resistors of R and 2R value (eg. 1K and 2K Ohm). Below, a 4-bit R2R DAC is exemplified. Bits b0 to b3 form the number to-be-converted into analog voltage (OUT). Assuming all R and 2R are of exact values, the table below. The currents from b0,b1,b2,b3 sum up into the 2R resistor to the GND: Vout = $Vb_0/16 + Vb_1/8 + Vb_2/4 + Vb_3/2$

Table 18. 4-bit R2R DAC output

b_3	b_2	b_1	b_0	Decimal value	OUT Voltage
0	0	0	0	0	0 * Vcc
0	0	0	1	1	1/16 Vcc
0	0	1	0	2	2/16 Vcc
0	0	1	1	3	3/16 Vcc
0	1	0	0	4	4/16 Vcc
0	1	0	1	5	5/16 Vcc
0	1	1	0	6	6 / 16 Vcc
0	1	1	1	7	7 / 16 Vcc
…	…	…	…		….
1	1	1	1	15	15 / 16 Vcc

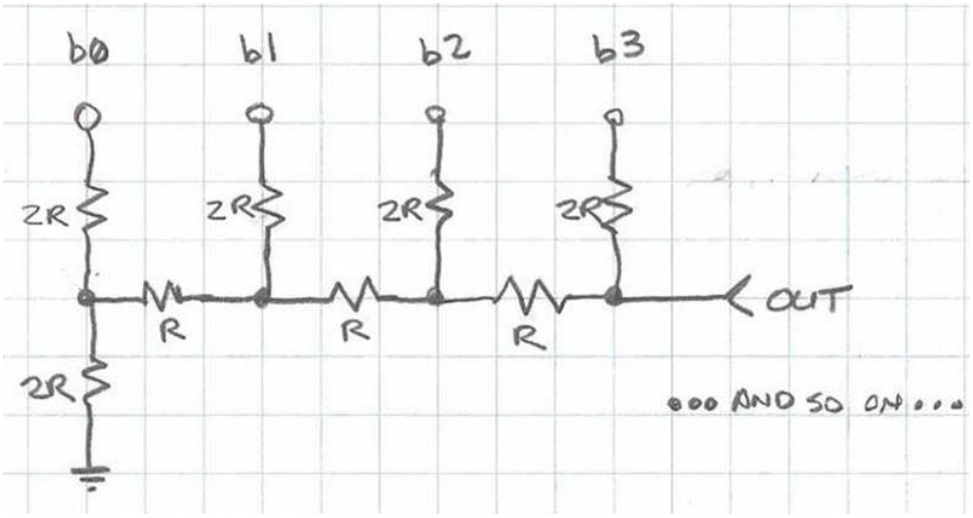


Figure 112 R2R DAC architecture

2.8.7 Resources: Comparators

The comparator is a high-gain differential amplifier which exacerbates any small difference between an input voltage and a voltage reference. The output is a digital signal (0 for less or 1 for more) that can be sent to a GPIO.

2.8.8 Resources: RAM memory

Although not a classical peripheral, some microcontrollers offer external interface to address external memory. This external memory may be used for programs (and is of type flash or EEPROM – Electrical Erasable Programmable Read Only Memory) or for data (and is usually of type SDRAM).

Even without external interface, microcontrollers do offer internal interface to their own program memory or data memory. The data memory is usually for type SRAM, whereas the program memory is flash based.

SRAM (static random-access memory) is more expensive than DRAM but offers higher speeds as it does not need periodic refresh like DRAM. DRAM memory is usually capacitor-based, so it can be very large and inexpensive, with the tradeoff of performance (needs periodic refresh, needs write after each read, has shared address row and column data lines etc.)

2.9 FPGAs

FPGA stands for Field Programmable Gate Array and, as the name says, they are programmable outside of the factory (in the field). They contain hardwired resources, tied together using a switch network, which can be reconfigured at runtime. This allows changing the hardware according to a specific software application, at runtime.

We will call a chip to be programmable if we can write a software (containing instructions hardcoded at the chip factory, during manufacture) and program that software in the program memory of that chip (MCUs) or in an external program memory chip (processors).

We will call a chip to be reconfigurable if we can reconfigure the internal connections inside the chip, to achieve various computing architectures, effectively changing the instructions/data paths/signal paths even if the chip was already manufactured and left the factory). These devices are called gate arrays (as they contain gates which programmable switches to link them), and usually are CPLDs or FPGAs.

Comon resources in FPGAs:

- LUTs (look-up tables), usually 4 -to-6 inputs and 1 or 2 outputs, that can store any data bits of 4-to-6 rows and 1-to-2 columns
- FF (flip-flops) are 1-bit storage units that store data when the clock line transitions
- DSP slices (ASIC circuits with hardwired computing, usually specialized in multiplications and additions)
- BRAMs (Block RAMs, spread over the area of the chip)
- A plethora (64 to 1000+) of I/O pins supporting various voltage and current standards

2.10 SOCs

These devices contain multiple chips in the same package (hence the name System on Chip) and are very useful in packing lots of processing power in a small physical form. They also have the advantage of linking components with very high-speed data buses.

2.10.1 SOCs (CPU + FPGA)

These devices contain single-core, dual-core or every quad-core CPUs together with an FPGA so that the programmable part is programmed by writing software and the reconfigurable part is programmed by writing a digital design.

An example of this chip is the Zynq 7000 family of chips, as depicted in Figure 113. The processing system is the CPU-part and the Programmable Logic is the FPGA-part.

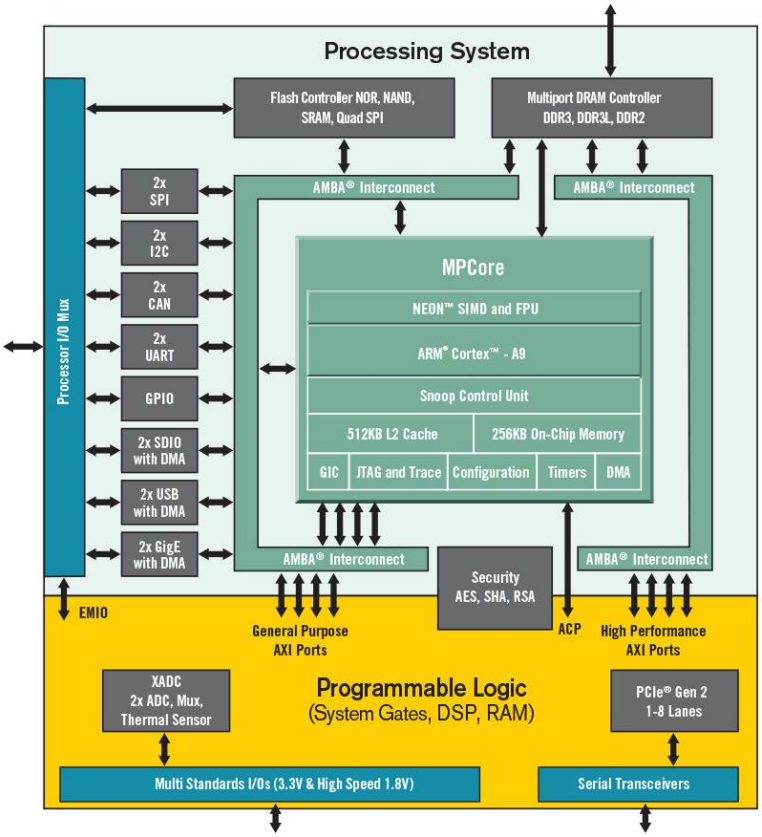


Figure 113. AMD/Xilinx Zynq 7000 SoC chip family

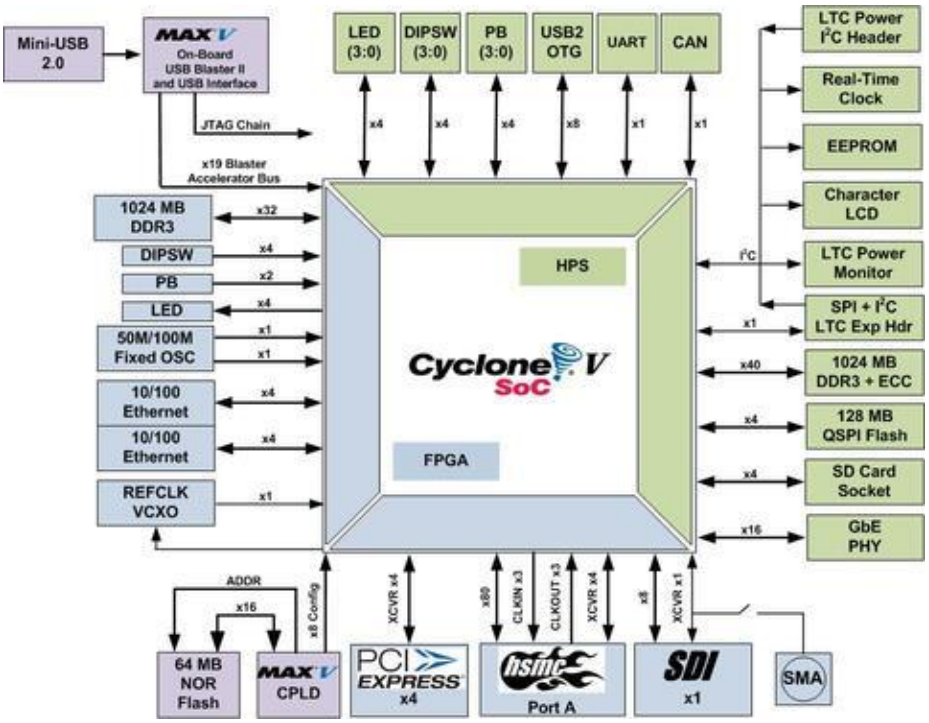


Figure 114. Intel/Altera Cyclone V SoC

2.10.2 SOCs (CPU + GPU)

These devices usually contain multicore CPUs (1-8) and manycore (usually 128 – 2048) embedded GPUs. An example of such chip is the one in Figure 115. In addition to the CPU cores, it offers a GPU with 8 symmetric multiprocessors, capable of running 1024 simultaneous threads, video decoding units and audio processors and a plethora of external interfaces.

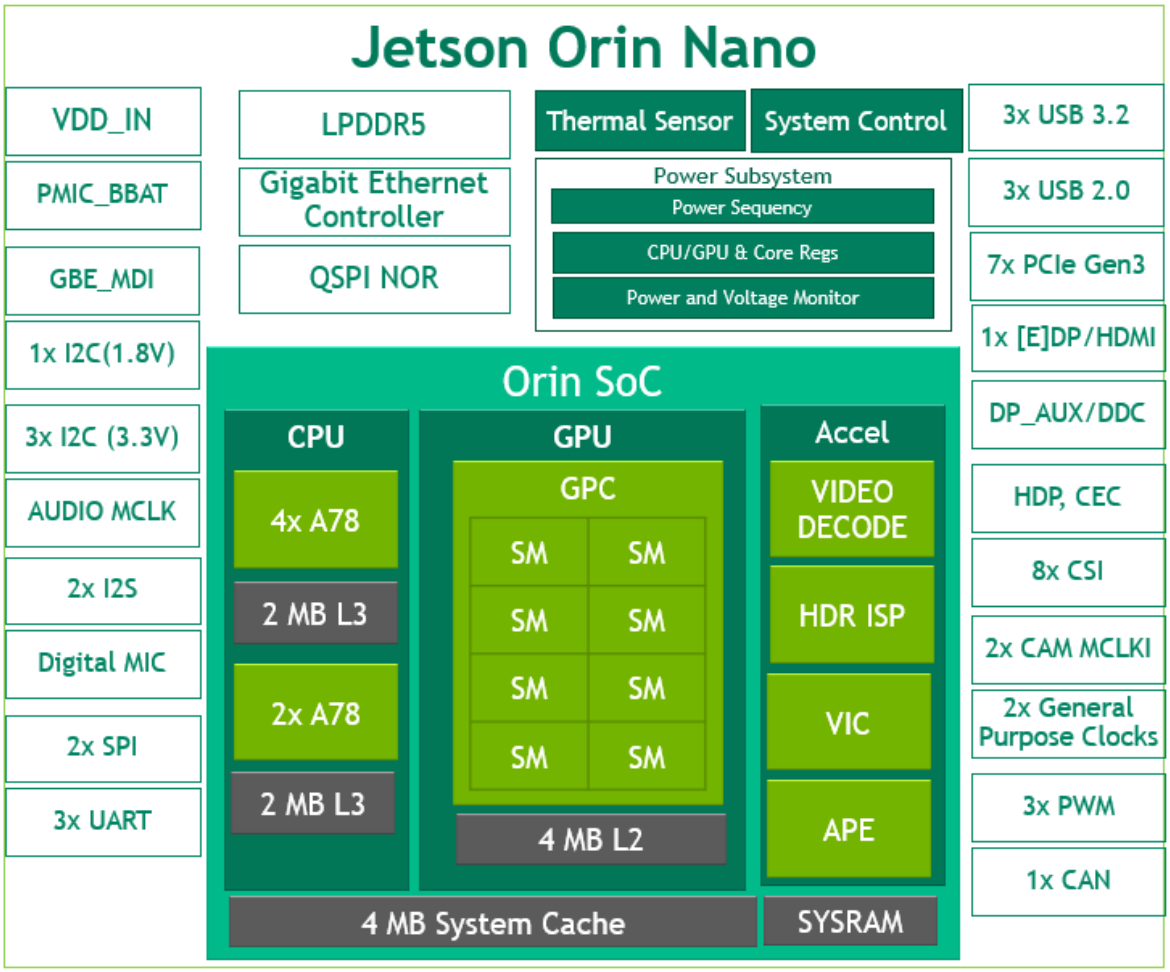


Figure 115. NVIDIA Jetson Nano SoC

2.11 Physical package of integrated circuits

To produce the digital circuits, the manufacturer must encapsulate the circuits in a protective package, usually ceramic-based. Different form factors and materials have different names, dimensions, and properties (resistance to humidity, vibrations, temperature etc.)

The 7400 series (1960s) of integrated circuits may still be bought (with their upgrade, the CMOS variant in 1980s) in common packaging formats like SOIC (small outline integrated circuit) or DIP (dual-inline package), and usually have a few gates of the same kind.

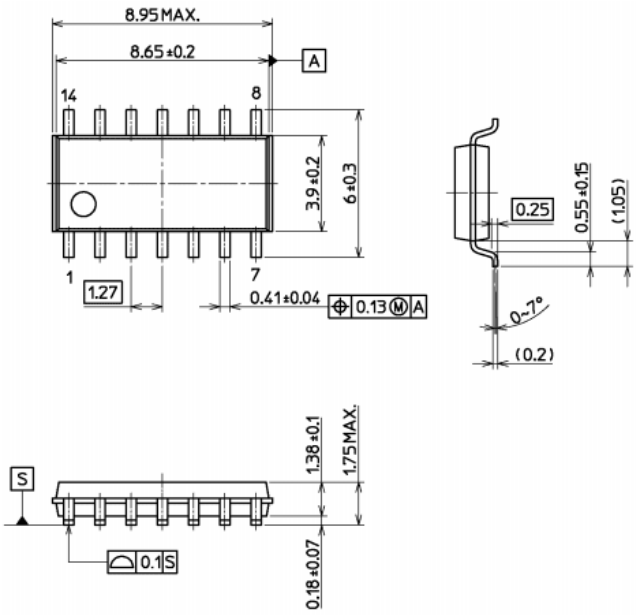


Figure 116. SOIC-14-N (narrow) package, having P (pin-to-pin distance) of 1.27mm (that is 0.05 inch, or 50 mils)

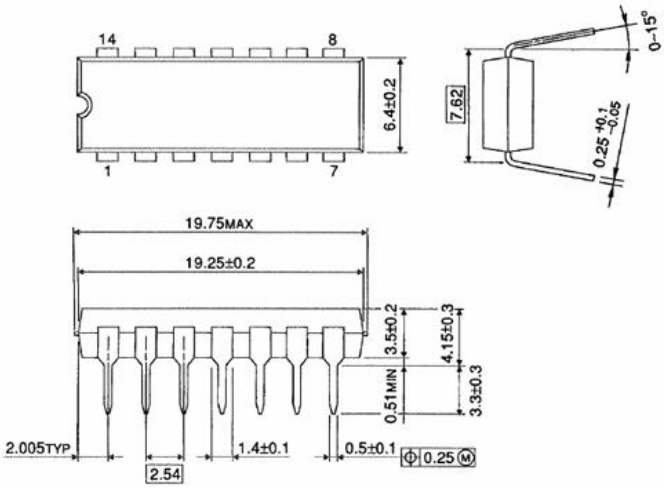


Figure 117. DIP14 package, having pin-to-pin distance of 2.54 mm (0.1 inch)

The SOIC (for SMD components) and DIP (for THT connectors) packages are preferred for manual soldering as they are small enough to have some PCB density, yet big enough to be visible to engineers' eyes without effort. Similarly, the 0805 package (length x width of 0.08 x 0.05 inch) for SMD passive components (resistors, capacitors) are preferred.

Integrated circuits are connected among themselves using a physical and electrical support called PCB (printed circuit board). The PCBs may contain multiple layers (most simple having 1 or 2 (named top and bottom), and most common and complex up to 16). Some layers are electrical, some are just for soldering or

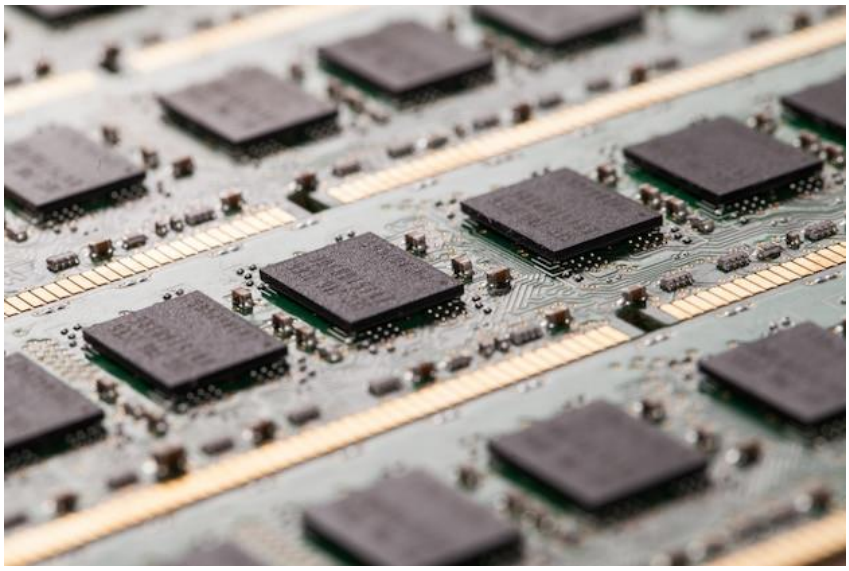


Figure 118. A PCB (printed circuit board) with multiple SMD components

There are multiple packages available for an electronic circuits, but based on the way it connects to the PCB (printed circuit board) they are SMD (surface mounted devices) or THT (through hole technology)

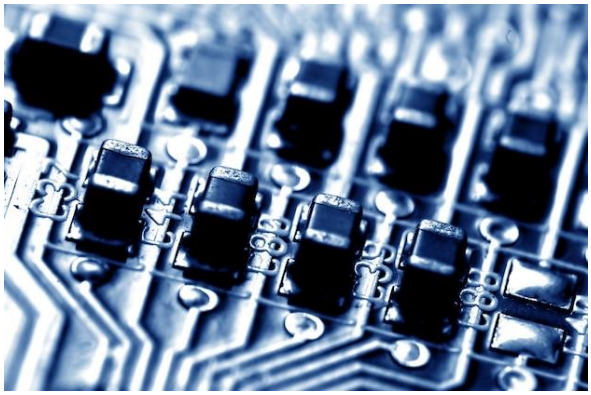


Figure 119. A PCB with SMD capacitors

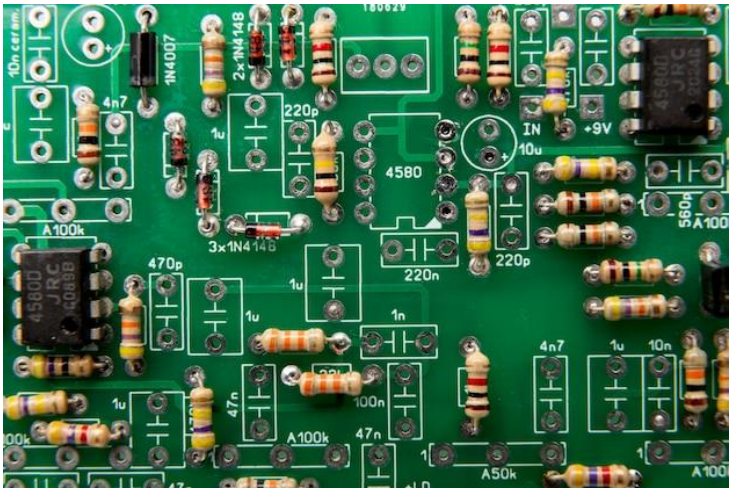


Figure 120. A PCB with multiple components in THT (diodes, resistors, ICs)

The PCs have as their main board (motherboard) a big PCB with multiple electrical layers, where thousands of components are mounted, usually in SMD technology. It relays all the electrical signals between main PC components: CPU (central processing unit), RAM memory, long-term storage (SSD/HDD), Video Card, I/Os (network interface card, USB ports, keyboard, mouse). A simplified layout is shown below:

Figure 121. A 3D rendering of a simplified PC motherboard.

The CPU is cooled with an active cooler (top), two PCIe slots (long, orange connectors), RJ45 connectors for networking (in a 2x2 packet, gray, middle), audio connector (blue, green, red stacked together: audio out, line out, microphone in), a common CR2032 BIOS-backup battery and two LEDs (bottom right, green and red)

2.12 Temperature ranges

Electronic equipment is made of multiple electronic parts. Depending on the market, one wants to be able to use the equipment in various climates, therefore all the components/parts must be able to withstand those temperatures. To ease component selection, manufacturers rate their products in “temperature ranges”.

Most common is the commercial temperature range. For a TV device, which is expected to be in your living room, will be subjected to a small temperature range. Multimedia units used in the cars, that can also play movies and run navigator apps, are expected to work in outside temperatures the car will be exposed to.

When TV manufacturers want to build a TV for the common living room, they will select components in commercial temperature range. When they want to build a TV screen for a car, they will select components in the automotive temperature range. The schematic might be the same, but the component type will change: usually an I or E or A at the end of the device name, denotes temperature range of Industrial, Extended Commercial or Automotive.

Table 19. Temperature ranges

Name	Temperature range (Celsius)
Commercial	0 to +70
Hi-Temperature Commercial	-10 to +100
Extended Commercial	-20 to +85
Industrial	-40 to +85
Automotive	-40 to +105
Full-Range Industrial	-40 to + 125
Power Supply	-40 to +130
Military	-55 to +125

3. Basic software bricks

3.1 Programming languages

Programming languages are tools for software development. They provide a human-readable way for programmers to express algorithms to computers and instruct them to how to perform tasks. The programming language is then compiled into machine-executable code, or interpreted, allowing developers to create high-level applications for nontechnical users. Each programming language has its own set of rules, syntax, and semantics, defining how code is structured and what operations it can perform. Some programming languages are general-purpose and versatile, while others are specialized for specific domains, such as web development, data analysis, embedded systems, high-performance computation, image processing etc.

Concepts of programming languages:

1. Syntax: The syntax of a programming language defines its grammar and rules for constructing valid statements and expressions. Syntax is crucial, as even minor mistakes (e.g., missing semicolon) can lead to errors that prevent the code from compiling (syntax errors) or executing (runtime errors) correctly.
2. Data Types: manipulated data may be integers, floating-point numbers, characters, strings, Booleans, arrays, and more. Each data type represents various kinds of information and may support different operations on it.
3. Variables and Memory Management: Variables are used to store and manipulate data during program execution. Variables are declared, initialized, assigned values to. In addition, memory allocation and deallocation can also be manually managed or automatic, depending on the language.
4. Control Structures: enable the flow of execution in a software program. They include conditional statements (if-else), loops (for, while), and branching (switch-case), and allow programmers to change the order of operations and make decisions based on specific conditions evaluated at runtime.
5. Functions and Classes/Modules: Functions are blocks of code that perform specific tasks and can be reused throughout the program (they are also called subprograms). The functions usually have names starting with a verb (because they do something) and should never be longer than a page. Multiple functions (behavior) and data are grouped into classes (in Object-Oriented Programming paradigm of more modern programming languages); they promote modularity and help break complex problems into smaller, manageable parts, enhancing code organization and readability.
6. Libraries and Frameworks: Programming languages often come with built-in libraries and support for external frameworks. These libraries and frameworks offer pre-

written code and functionalities that developers can utilize to speed up application development. For example, OpenCV [14] is a library for image-processing, AngularJS[15] for web-frontend, Django[16] (Python-based webapp development), Express [17] (Node.js backend for web apps)

Popular Programming Languages:

There are numerous programming languages available, each with its strengths and weaknesses, suited for various domains. Top six most widely used programming languages are:

- C/C++: These languages are valued for their performance and are commonly used in system programming, embedded systems, and game development. Unfortunately, there is a steep learning curve for using them.
- Python: Known for its simplicity, is popular for web development, data analysis, artificial intelligence, and scripting tasks. It rose to prominence as it allows the programmer to do a lot in one line of code (as it has a lot of libraries for processing data, reading/writing images, movies, running inference or training on neural networks etc.)
- Java: Java is a versatile language, widely used in enterprise applications, Android app development, and web services. It is a strongly typed language.
- C#: Developed by Microsoft, C# is widely used for Windows application development and game development using Unity. Its syntax resembles most to Java.
- JavaScript: Primarily used for web development, JavaScript allows interactive and dynamic functionality in web pages. It is both used for front-end and back-end.

Below is a sample of a “Hello World” program (a simple program that lists at the console / on the monitor in text mode, a string of those two words) written in those six languages:

Table 20. Hello world written in C and C++ language.

C	**C++**

#include <stdio.h>
int main() {
printf("Hello World");
return 0;
}
	#include <iostream>
	int main() {
	std::cout << "Hello World";
	return 0;
	}

Table 21. Hello world written in C# and Java language.

C#	Java

namespace HelloWorld
{
  class Hello {
  static void Main(string[] args)
  {
  System.Console.WriteLine("Hello World");
  }
  }
}
	import java.io.*;
	
	class Hello {
	  public static void main (String[] args) {
	  System.out.println("Hello World");
	  }
	}

Table 22. Hello world written in JavaScript and Python language.

JavaScript	Python
console.log("Hello World");	print("Hello World")

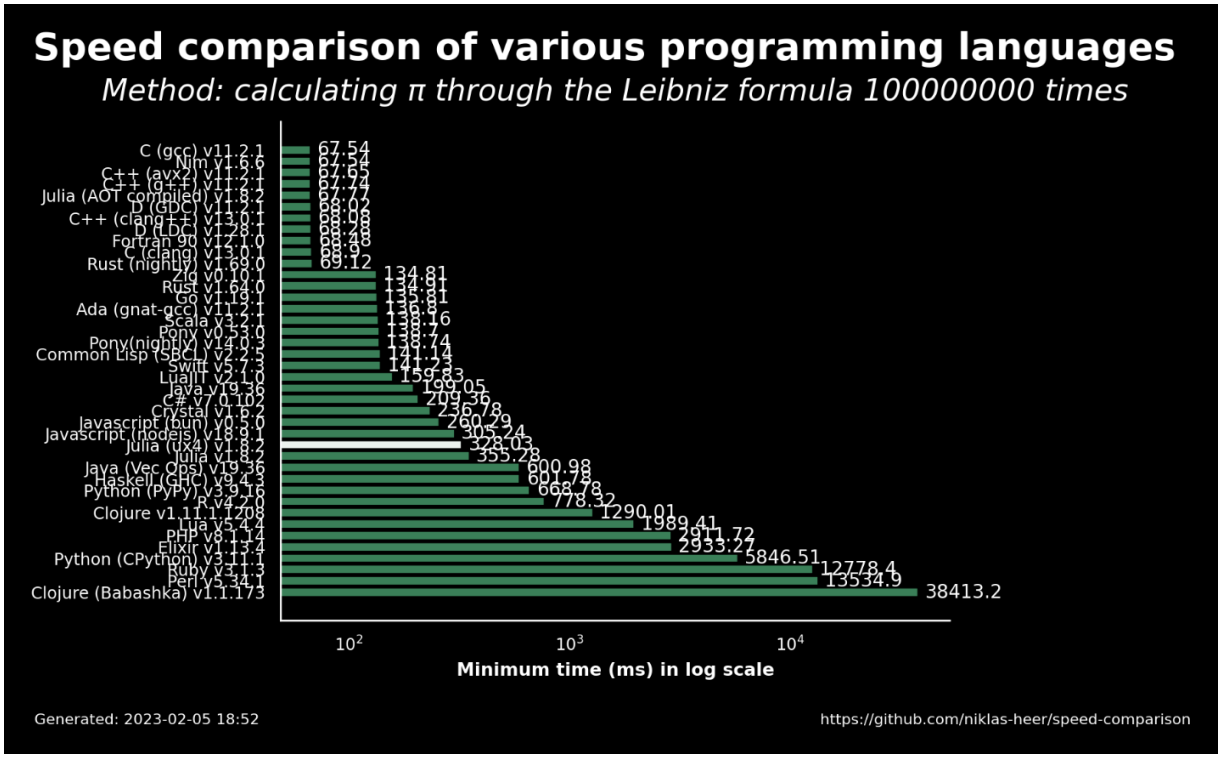


Figure 122. A comparison of speed of intensive computation across multiple languages. C/C++ is fastest when compiled with optimizing compiler (gcc)

3.2 Variables and constants

A constant is a well-defined number which does not change its value during the run of the program. A variable is a memory location that may change its value by assignment. The name of such object are usually made of letters, commonly nouns (most languages prevent defining a name starting with a number).

When a constant or variable is defined, most languages require a type (or assume one). The types of variables/constants will influence the amount of space it will occupy in the memory, and the amount of data it will be able to store.

In C/C++ language, most common variable types are:

Table 23. Most common variable types in C/C++ languages.

Old name (C89)	**New name (C99)**	**Possible values**	**Description**	**Constant number-to-number gap**
char unsigned char	int8_t uint8_t	-128 to +127 0 to _255	One alphanumeric character	YES
short int unsigned short int	int16_t uint16_t	-32768 to +32767 0 to + 65536	An integer number	YES
long int unsigned long int	int32_t uint32_t	-2,147,483,648 to +2,147,483,647 0 to +4,294,967,295	A larger integer number (for 32-bit systems)	YES
long int	int64_t	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	A larger integer number (for 64-bit systems)	YES
float	float	-3.402823466 E + 38 +3.402823466 E + 38	A simple (32-bit) precision floating-point number	NO
double	double	-1.7976931348623158 E + 308 +1.7976931348623158 E + 308	A double (64-bit) precision floating-point number)	NO

NB: E+38 means 10 to the power of 38.

Table 24. TIOBE index [13] for programming languages

Aug 2023	Programming Language		Ratings
1		Python	13.33%
2		C	11.41%
3		C++	10.63%
4		Java	10.33%
5		C#	7.04%
6		JavaScript	3.29%
7		Visual Basic	2.63%
8		SQL	1.53%
9		Assembly language	1.34%
10		PHP	1.27%
11		Scratch	1.22%
12		Go	1.16%
13		MATLAB	1.05%
14		Fortran	1.03%
15		COBOL	0.96%
16		R	0.92%
17		Ruby	0.91%
18		Swift	0.90%
19		Rust	0.89%
20		Julia	0.85%

3.3. Operators

The usual operators applied to numbers are in most languages, inherited from mathematics. A table of such common operators may be found in the table below:

Table 25. Common operators in languages

Symbol	Explanation	Type of operator
! ~ & | ^	logical negation bitwise negation bitwise AND bitwise OR reduction XOR	logical bitwise bitwise bitwise bitwise
&&	logical AND	logical
||	logical OR	logical
* / %	multiply divide modulus	arithmetic arithmetic arithmetic
+ -	binary plus binary minus	arithmetic arithmetic
<< >>	shift left shift right	shift shift
> >= < <=	greater than greater than or equal to less than less than or equal to	relational relational relational relational
== !=	logical equality logical inequality	equality equality
?:	if-the-else	conditional

Note: Most languages add their own operators with special meaning, so be sure to check the standard before coding into one of them.

Common standards for **C language** are: C89, C90, C99 [7], C11, C17;
Common standards for **C++ language** are C++98, C++03, C++11, C++14, C++17
Common standards for **VHDL language** are IEEE 1076-1987, IEEE 1076-1993, IEEE 1076-2000, IEEE 1076-2002, IEEE 1076c-2007, IEEE 1076-2008, IEEE 1076-2019
Common standards for **Verilog language** are: IEEE Standard 1364-1995 (Verilog-95), IEEE Standard 1364-2001 (Verilog 2001), IEEE Standard 1364-2005 (Verilog 2005), and later, with the addition of the superset System Verilog for verification and design modelling, SystemVerilog 2009 (IEEE Standard 1800-2009) and later, 2017.

4. Complexity

Complexity may be classified as:

- Complexity of behavior
- Complexity of structure

The complexity of an algorithm: the usual meaning is the manner execution time increases with data size, or the manner the memory consumption increases with the data size.

The complexity of a circuit: the usual meaning is the manner the number of gates increases with number of inputs or size of inputs.

Complexity is usually expressed as Big-O notation. The complexity of a function f is decided by finding another function g, which asymptotically bounds the f function.

Mathematically, this is expressed as: if f(n) has the same complexity as g(n), then from n = k onwards, the c*g(n) >= f(n), where k is a point < infinity and c is a constant

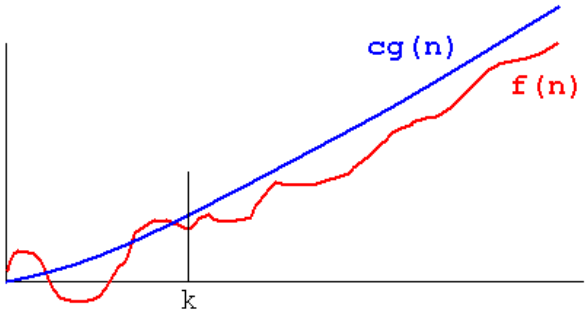


Figure 123. Representing a g(n) function which asymptotically dominate f(n) function.

To learn more about Big-Oh, Big-Theta and Big-Omega and their small variants, see [8]

A list of common complexity classes is listed below. O(1) means constant time, and is the best one can hope for: the algorithm runtime will not increase with data increase, the circuit will not increase its size no matter the number of inputs etc.

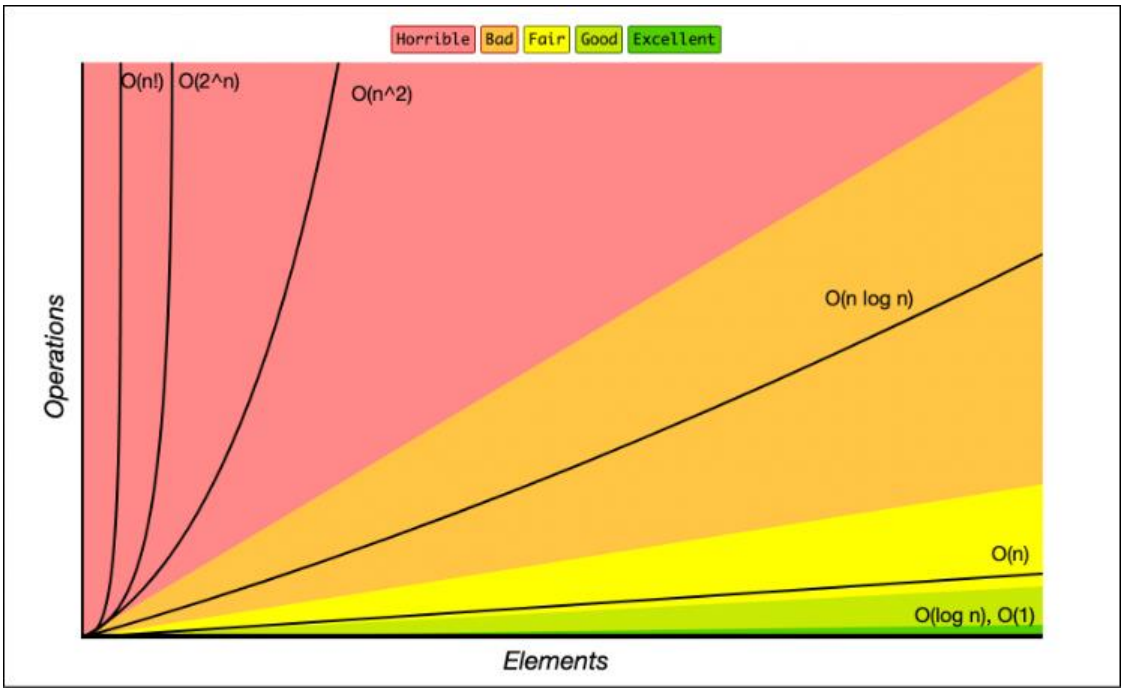


*Figure 124. The O-notation complexity increase with the number of elements processed. O(1) and O(logn) are usually excellent complexities, O(n) is fair, and O(n*logn) is usually considered almost decent in algorithms and circuits*

To better understand the complexity difference between O(logn) and O(n) we propose the next game: one thinks at a number from 0 to 100. Assuming another one tries to guess the number, with hints of "my number is higher" or "my number is lower", in O(n) algorithm (bruteforcing all values) one will guess in at most 100 steps, whereas in O(logn) algorithm, same task will take at most 7 steps (assuming log2). This not very impressive, but as one goes towards infinity, the advantage will become obvious. Assume the same game, but with numbers from 0 to 4 billions. In O(n) it will take at most 4 billions steps, whereas in O(logn) it will take at most 32 steps! This is the power of the logarithm.

4.1 Examples of algorithms and their complexities

Ex1: Sorting a vector of numbers takes at least O(N) - one has to go through all the vector at least once, but the worst case for best algorithm known, is O(NlogN).

Ex2: Finding a minimum value in an unsorted vector will take:

- By going through all the values of the vector once, O(N)
- By sorting the vector, then choosing the first value, O(NlogN). One could argue that this method is worse than the first one, and, if only one minimum value is needed, it would be true. However, if at least logN minimum values must be chosen, this methos of sorting might be the better one.

Ex3: Matrix-matrix multiplication

What is the complexity of multiplying A matrix of size MxK with B matrix of size KxN into a resulting matrix C of size MxN ?

Assume for simplicity that all matrices are NxN. The obvious absolute minimum amount of multiplications is N*N (you still have to sweep through all the lines and columns); However the best known algorithm so far (2023, see [9]) is O(N** 2.371552). The Strassen algorithm gives a O(N**log2(7)) that is approx. O(N**2.807), whereas the naïve approach is O(N**3).

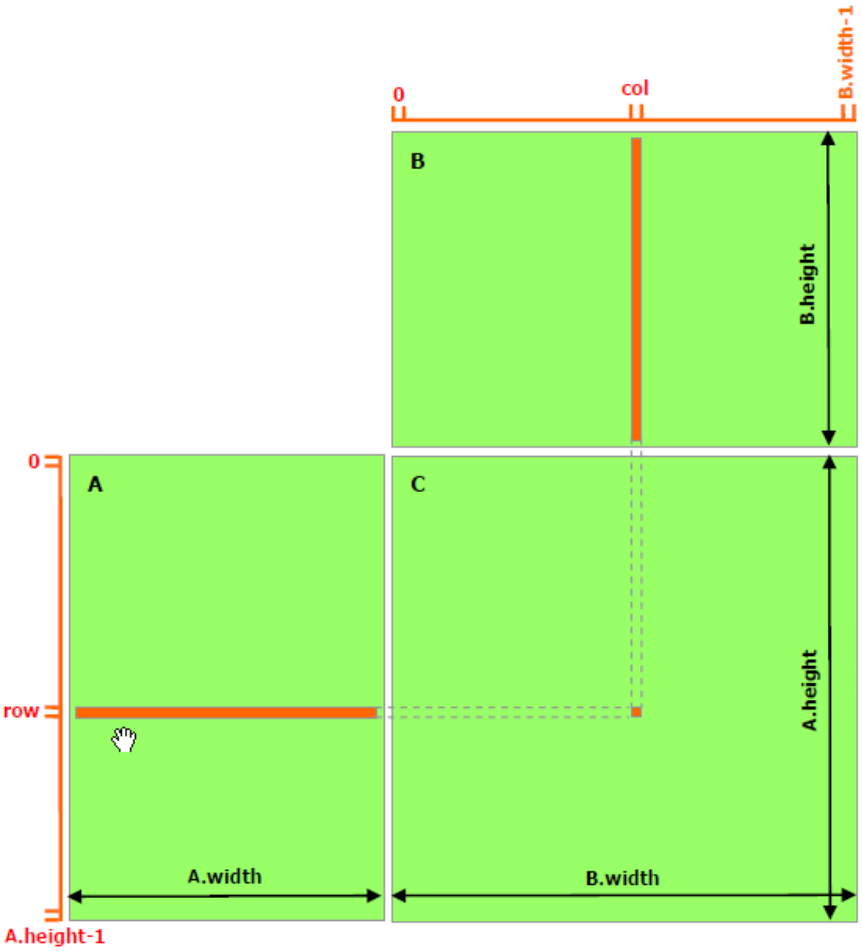


Figure 125. Matrix-matrix multiplication (naïve-approach)

Table 26. An example of algorithm complexity impact on running time, as a function of data size (green).

Data size	K	1	10	100	1000	10000	100000
Complexity	Runtime in e.g. microseconds = f (data size)						
O(1)	10	10	10	**10**	10	10	10
O(logN)	5	0	5	**10**	15	20	25
O(N)	2	2	20	**200**	2000	20000	200000
O(NlogN)	1	0	10	**200**	3000	40000	500000
O(N**2)	0.5	0.5	**500**	50000	5000000	5.0E+08	5.0E+010
O(N**3)	0.02	0.02	**500**	500000	5.0E+08	5.0E+011	5.0E+014
O(2**N)	0.01	0.02	1.024	1.26E+027	1.07E+298	Too large	Too large

In the above table, the multiplying constant is intentionally decreasing to show that for a small amount of data, higher-complexity algorithms might still do the job faster. In this example, for running with 1 data, the fastest algorithm is the O**3 and O(2**N); however, as data size increases, only O(1) and O(logN) have small running times.

4.2 Rules in O-notation

4.2.1 Product

4.2.1.1 Product rule #1 Given f1 = O(g1) and f2 = O (g2) ⇒ f1 * f2 = O(g1 * g2)

4.2.1.2 Product rule #2 f*O(g)=O(f*g)

4.2.2 Sum

If f1 = O(g1) and f2 = O(g2) then f1 + f2 = O(max (g1,g2)). The interpretation of this rule is that the function which increases faster to infinity, decides what complexity the sum is.

4.2.3 Multiplication by a constant

Let k be a nonzero constant. Then O(|k|·g) = O(g). In other words, if f=O(g), then k*f = O(g). We will say that "constants do not matter when going to infinity." This is an expected behavior derived from the definition of the Big-Oh notation:

*If f(n) has the same complexity as g(n), then from n = n0 onwards, the c*g(n) >= f(n), where n0 is a point < infinity and c is a constant*

The impact of such a k constant, would only be to push the c constant to a higer value, without changing the way g(n) dominates f(n).

4.3 Other considerations involving Big-Oh notation

Why all logarithms are equal when we talk about complexity?

In the above example, one can notice that the author wrote logN, without specifying the base of the logarithm. This is intentional, as the base of the logarithm does not matter when discussing about Big-Oh. To prove this, let us remember the change-of-base for the logarithm:

$$\log_a(b) = \log_c(b) / \log_c(a)$$

Say a = 2 and c = 10, then $\log_2(b) = \log_{10}(b) / \log_{10}(2)$, so the difference between a $\log_2(b)$ and a $\log_{10}(b)$ is only a constant (in this case, $1/\log_{10}(2)$ = approx. 0.6).

Since multiplication with constant has no impact on the O-complexity, one does not have to specify explicitly the base of the logarithm. Another interpretation would be that all logarithms increase their value towards infinity, in the same way (with same speed).

Why are not all problems created equal?

P vs NP is an open problem, it mainly states that: assuming a solution to a problem may be checked quickly (in polynomial time, O(N)), is there a way to find that solution equally fast (in O(N)) ? If this is true, it would have great implications in a lot of domains, for example in cryptography, where it is extremely easy to check if a decryption key is the right one, but it is extremely hard (usually exponentially hard) to find it.

This P vs NP is one of the seven Millenium Prize Problems (where Clay Mathematics Institute awards 1 million dollars for each solved problem). Only one such problem was proven (the Pointcarré conjecture), and the Russian mathematician that did it, Grigori Yakovlevich Perelman, declined the award in 2010.

4.4 Why does encryption (currently) work?

4.4.1 What is encryption?

Encryption is a way to convert information seen as unencrypted data (“plaintext”) into secret code (“cyphertext”) that hides the information’s true meaning. The science of encryption and decrypting information is called cryptography. Cryptography was used since ancient times (1900 BC, in Egypt) and later on, in Greece and Roman military.

The Caesar cipher is famous for posing problems to the uninitiated of that time (although now it is easily breakable by any student). It relies on replacing any letter in the alphabet (A, B, C.. with their equivalent letter situated at a constant distance in the alphabet eg. D,E, F... assuming the distance is 3). As a hit on how to brake it: brute forcing will only require at most 26 tryouts. This is a hint on how to make a code resistant to breaking: make brute-forcing unfeasibly long; as one shall see it is not the only thing that matters, as some faster way to break it, might still be possible.

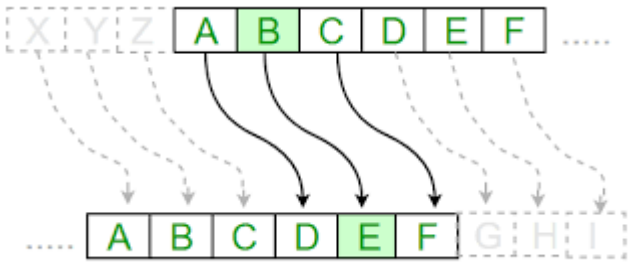


Figure 126. Caesar cipher exemplified, for a distance of 3

One of the problems of the Caesar cipher is that each letter is always encoded with the same output letter, thus the language statistics (vowels are more used than consonants + each letter has its own frequency in frequently-used words) will be its downfall.

The Vigenère cipher is an evolution of Caesar cipher, where each letter is replaced by another letter in the alphabet in a key-dependent manner (the equivalent of a Caesar cipher for each letter in the plaintext, the distance of such ciphers is given by specific letter in the key used to encipher). For about 300 years from 1553 to 1854 it was thought to be unbreakable.

The amount of combinations increases to 26**key_size. Assuming key_size is 1, it is the same as Caesar cipher. But assuming the key_size is 26, then the number of combinations to try by brute force, is about 6*10**36, which even today, on the most powerful TOP500 supercomputer (see [11]) would take about 194 billion years! As one can see, even today, assuming the key is long enough (and not an English word) and its length is not known, the

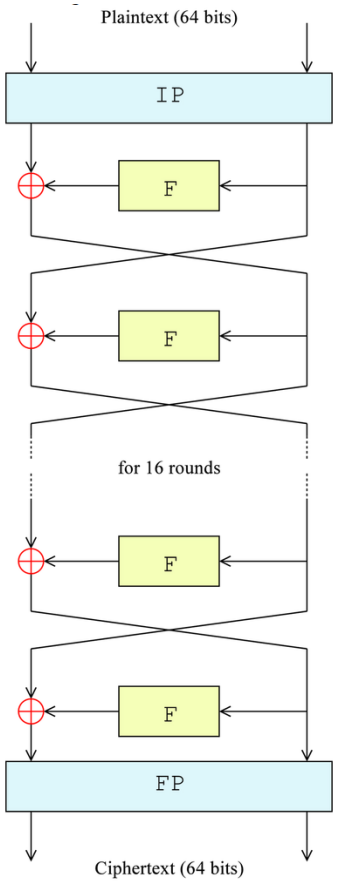
Vigenère cipher is still top notch. This is because of the OTP (one time pad) is unbreakable: the OTP relies on the following four properties:

- The key must be at least as long as the plaintext
- The key must be random (uniform distribution, independent on the plaintext)
- The key must never be reused (not even in part)
- The key must be secret

If such a code exists, why doesn't everyone use it? It is because one has to share the key, secretly between the sender and the receiver; this is not always possible, sometimes the transmission medium used, is unsecured (e.g. air, long electric wires, long fiberoptics). What people usually use, is a key-sharing (key-exchange) algorithm (that takes a long time to crack) and then replacing it from time to time, to transfer high amount of data. The idea is to secure some packets, that even if they become known at some point in the future, that would be useless, as they became obsolete...

4.4.2 Symmetric-key cryptography

This type of encryption/decryption uses the same key for both encryption and decryption.



Symmetric-key algorithms are either stream ciphers (Caesar) or block ciphers (DES, 3DES, AES). DES (standardized in FIPS46 []) is based on Feistel network (using a one-way function), which is used for 16 rounds. The data is split into blocks of 64 bits and at each round, half of the data goes unaffected, and half goes through the substitution boxes (S-boxes).

After a key schedule which expands the 56-bit key into multiple 48-bit subkeys, and an initial permutation, 8 substitution boxes are used to replace the bits in a half block (32-bits) for each round. A final permutation is done in the end, and the ciphertext is output.

Figure 127. DES encryption where F is a Feistel one-way function

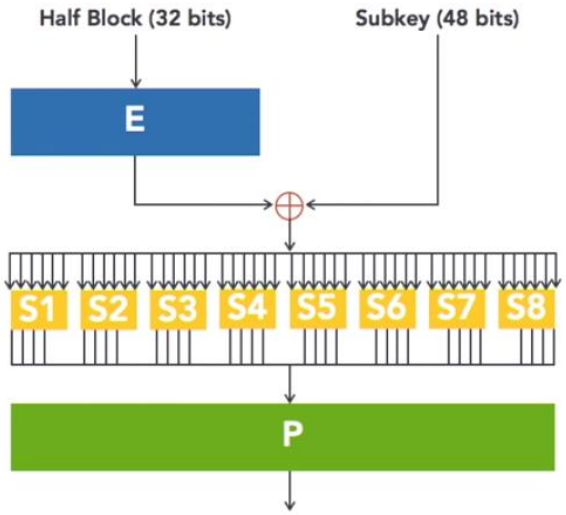


Figure 128. One round of DES encryption, Sx are the 6-bit to 4-bit substitution boxes

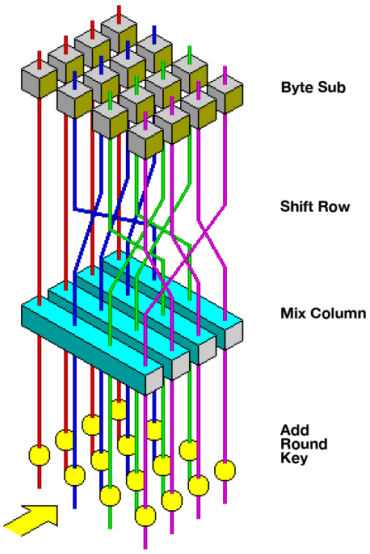


Figure 129. One AES round, expressed graphically in 3D

AES (FIPS 197 [20]) is based on a substitution-permutation network. The plaintext is split into blocks of 16 bytes, that are arranged in a 4x4 matrix ("state matrix") which is processed using the following operations to encrypt:

- Key expansion (generating round keys from AES-key, according to key schedule)
- AddRoundKey (XOR between round key and plaintext bytes)
- Multiple rounds of (10 for AES-128) :
    - SubBytes (a substitution for each byte using a lookup table)
    - ShiftRows (a rotation of rows in the state matrix)
    - MixColumns (a linear transformation of columns of state matrix)
    - AddRoundKey
- Final round with:
    - SubBytes
    - ShiftRows
    - AddRoundKey

The most used flavours of AES are AES-128, AES-192 and AES-256 (depending on key size). The state matrix is always 128-bit (4x4 = 16 bytes)

Bruteforcing AES-128 would take about 2158 billion years at the current rate of bitcoin network. It may be improved by a factor of 4, but it will require 72 PB of storage.

In block ciphers, the plaintext is split in multiple blocks of same size, and each block is encrypted *separately*: the way the blocks assemble into the cryptotext is called "chaining

modes" or "mode of operation" and were first standardized by FIPS81 [12] / DES modes of operation (ECB, CBC, OFB, CFB modes). In 2001, with the proposal of AES block cipher, CTR mode was added. If data dependency between blocks occurs, then parallel processing of multiple blocks is impossible.

Not all modes are created equal, some are to be avoided as the same block is encrypted in the same way always (see Figure 19 for a comparison), and some are better suited for parallel processing while still retaining great resistance to *cryptanalysis*.

Cryptanalysis is the process of analyzing encryption systems to understand resistance to attacks. There are multiple ways of doing this: differential cryptanalysis, meet-in-the-middle, related-key, etc. Since this is an ongoing process, never use a system that was not under review/attacks for multiple years. In addition, history is full of "security by obscurity" schemes, which were breached at some point and produced high losses (see AACS cryptographic key unveiling)

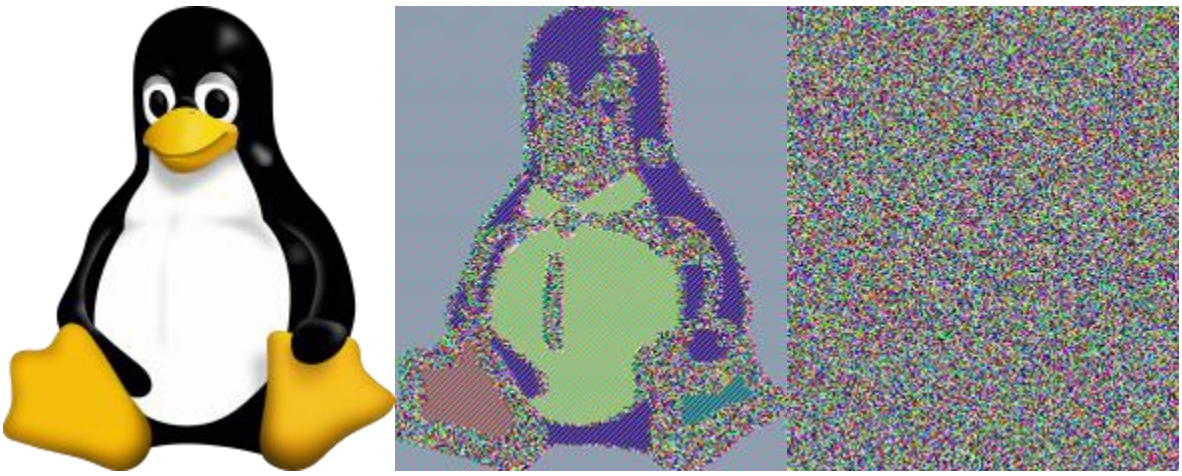
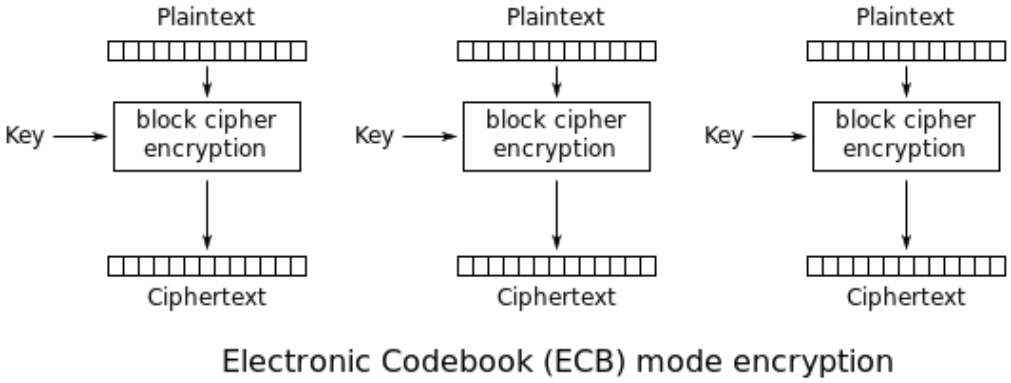


Figure 130. Left is the original image, middle is ECB mode cryptotext and right is CTR



Electronic Codebook (ECB) mode encryption

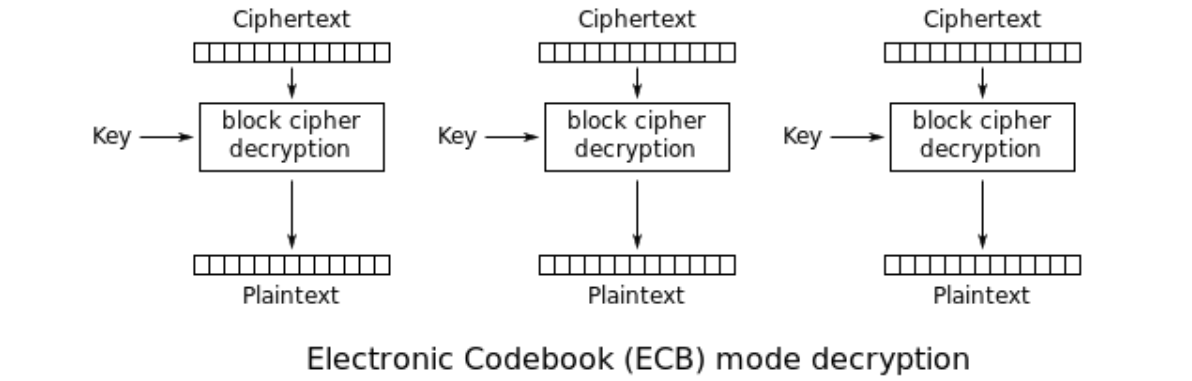


Figure 131. Top is ECB encryption and bottom is ECB description: great for parallel processing but very weak in hiding data.

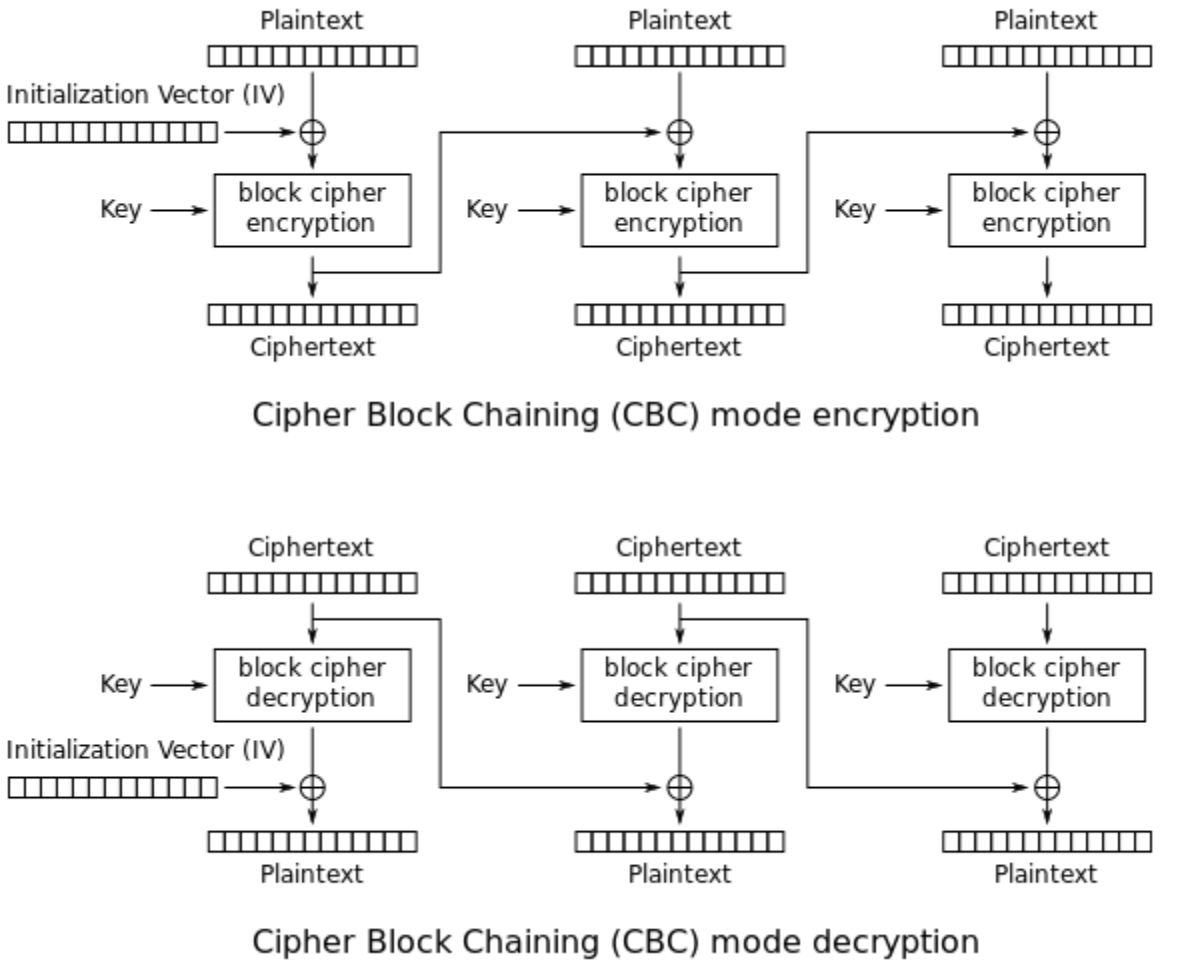


Figure 132. Top is CBC encryption and bottom is CBC decryption. CBC is hard to parallelize in encryption, but it is suitable for parallel decryption

Other modes exist like the OFB (output feedback mode) where there is data dependency on the encryption and decryption (previous encrypted text is at the input at the next encryption), or CFB (cipher feedback mode) where decryption is parallelizable.

Table 27. Block ciphers mode of operation with their attributes regarding parallel processing and strength

Mode of operation	Parallel encryption	Parallel decryption	Random read access	Weak correlation between plaintext and cryptotext
ECB	YES	YES	YES	NO
CBC	NO	YES	YES	YES
PCBC	NO	NO	NO	YES
CFB	NO	YES	YES	YES
OFB	NO	NO	NO	YES
CTR	YES	YES	YES	YES

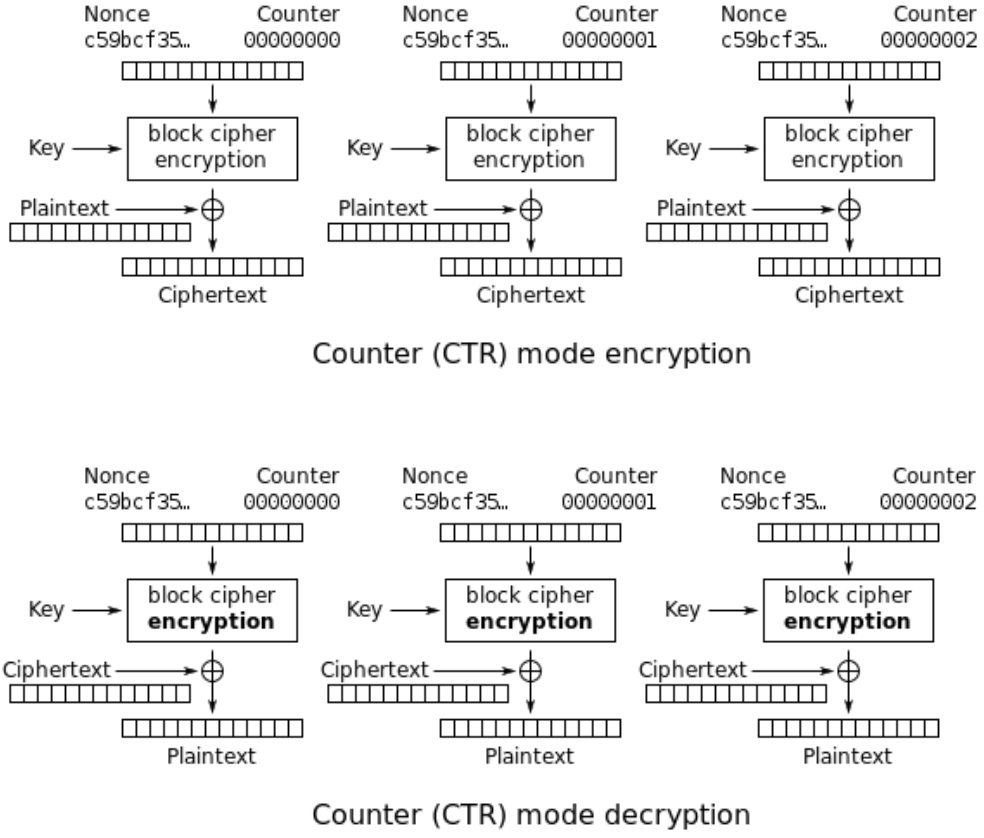


Figure 133. CTR mode encryption (top) and decryption (bottom).

CTR mode is the best mode to be used for parallel processing as there is no apparent data dependency among blocks.

4.4.3 Public-key cryptography

This type of encryption/decryption (also named asymmetric cryptography) uses different keys for encryption and decryption: the public key is used for encrypting (so that everyone can send an encrypted message to you) but then it uses a private key for decrypting (so that only the intended recipient may be able to decrypt it).

Most known public-key cryptographic schemes are ECC-based (Elliptic Curve Cryptography) and RSA-based (Rivest-Shamir-Adleman)

RSA is based on modular arithmetic; the main idea is the difficulty to factorize large integers. The public key is the multiplication of two large prime numbers and the private key is derived from the same two prime numbers. Key size gives the strength to the code.

Algorithm for setting up RSA encryption and decryption:

- Select two prime integers p and q. Compute n = p*q and Q(n) = (p-1)*(q-1). This is the number of integers less than n, that are relative prime with n. For example Q(12) = 4, as the 1,5,7,11 are relative prime to 12. Fermat's Little Theorem states that if p is prime and p does not divide x, then $x^{p-1} = 1 \bmod p$. The generalization of this theorem, called Euler's theorem, states that if x is relative prime to n, then $x^{Q(n)} = 1 \bmod n$
- Choose an encryption key: get e prime to Q(n). Public key is the pair (e,n)
- Calculate decryption key d such that $e*d = 1 \bmod Q(n)$
- Make the e and n public, keeping d (secret key) p and q secret.

Assume A wants to send a message to B. A will use the public key of B, encrypt with it, and then B will used his own secret (private) key to decrypt the message. Let us see how (and why) the encryption and decryption works:

- Encryption (A sends the message to B): get the public key of B and encrypt (compute M^e) $C = (M^e \% n)$, where % is modulo operation (reminder of division)
- Decryption (B receives the msg): get the private key of B and decrypt (compute C^d) $C^d = (M^e \% n)^d = (M^{ed-1+1} \% n) = [(M^{ed-1}) * (M^1)] \% n = M^1 \% n = M$, which proves the decryption process.

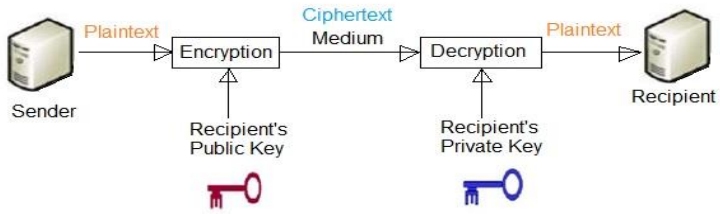


Figure 134 Public-key cryptography (encryption and decryption)

5 Mixed-signal application design

A lot of platforms exist with the purpose of helping the engineer design and implement a mixed-signal application.

One of the most common and affordable microcontroller platforms for implementation is Arduino platform, which allows C++ programming over a large array of embedded devices in a single-thread application. Its advantage is the large support across devices and sensors, due to its large size of implemented libraries. Pretty much any platform that allows development in C++ using a compiler, may be ported to Arduino by writing a "Arduino core support" targeting that platform. The Arduino company created an IDE for easy code development and also manufactured their own microcontroller kits based on AVR or SAMD microcontroller series.

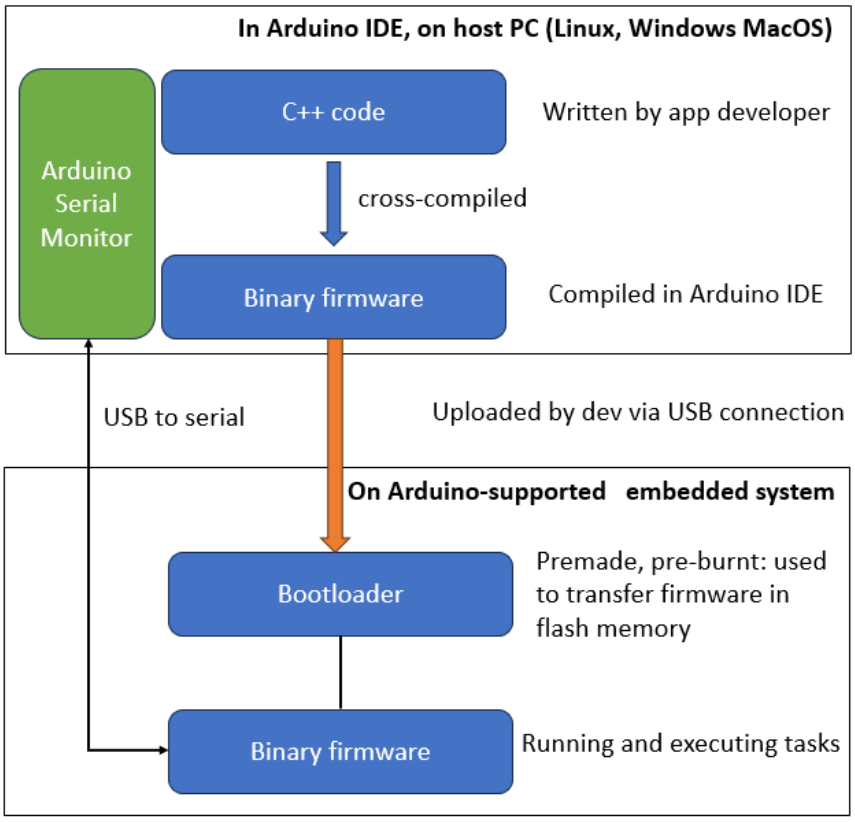


Figure 135 Arduino environment for developing apps

Other embedded platforms for development integrate both a CPU and a programmable logic array (usually FPGA – Field Programmable Gate Array) or a CPU and a GPU (like NVIDIA Jetson or AMD Kria).

5.1 Sensors for data acquisition

In digital or analog systems, sensors are devices that convert non-electric data into electric data, with the purpose of gathering data from the environment and allowing a processing system to respond to changes.

Depending on the type of the sensors: passive sensors are the ones that do not require electrical power to function, whereas the active sensors are the ones that need to be powered to function.

Depending on the type of the output, the sensors are analog (output is a voltage or a current) or digital (output is done through a digital bus/ data protocol)

Sensor type	From	To	Type	Example (model name)
Thermal infrared	Temperature	I2C data	Digital / active	MLX90614, non-contact infrared thermometer
Thermal	Temperature	I2C data	Digital / active	AMG8833, 8x8 pixels
Thermal	Temperature	1 Wire data	Digital / active	DS18B20, digital thermometer
Thermal	Temperature	Resistance	Analog / passive	K276/12KOhm / NTC
Thermal	Temperature	Resistance	Analog / passive	C1451 PTC 25…7500
Light	Infrared light	Open collector, voltage	Analog* (on/off)	RPM1738, for infrared remote control receivers
Light	LASER	Push-pull, voltage	Analog* (on/off)	CL3409, for virtual barrier
Light	Visible light	Resistance	Analog / passive	GL5528 / 12mm photoresistor
Light	UV Light	Resistance	Analog / passive	FDS010 photodiode
Light	Infrared	Voltage	Analog* (on/off)	PIR BS412, movement detector
Ultrasonic	Sound	Time delay, voltage	Analog* (on/off) delayed, active	HC-SR04, for distance measurement
Magnetic	Magnetic field	Voltage	Analog* (on/off), active	MLX92213, contactless encoders
Sonic	Sound	Voltage	Analog, active (for amplifier)	Grove-Sound Sensor based on electret microphone
Acceleration	Acceleration Magnetic field	I2C or SPI data	Digital / active	MPU9250, gyroscope, accelerometer and magnetometer
Barometric	Air pressure	I2C or SPI data	Digital / active	BMP280 pressure sensor
Force sensing	Mechanical pressure	Resistance	Analog / passive	FSR402
Humidity	Air humidity	I2C data	Digital / active	DHT11 humidity sensor

Sensors classified as analog*, output data as a level of HIGH or LOW, without and data encoding or protocol; in this regard, one might say the sensor has digital output…

5.2 Systems for data processing/storing

Embedded systems are systems where the computing machine (a microcontroller or microprocessor) is included as integral part of the system. They do not usually have a GUI, and files or operating systems are tightly enclosed in themselves.

High-performance clusters are systems comprising multiple computer servers tightly coupled and programmed together so that they can be viewed as a single system; they have fast, low-latency network connection for data exchange, large memories, and compute power.

Computers are systems in-between the embedded systems and clusters; they have the advantage of allowing a large customization option to fit almost all common tasks: video processing, video editing, photo editing, music editing, music playing, data storage etc.

5.2.1 Embedded systems

An example of such an embedded system is a NAS storage device, which has a microprocessor, a mainboard, SATA controllers for the drives and a network interface. The GUI, if it is provided, is as remote desktop connection (they do not have a monitor where the user can see operational parameters or issue commands).

Figure 136 Synology DS223j NAS unit

Some specifications of the Synology NAS unit are:

- CPU Realtek RTD1619B
- Memory 1 GB DDR4
- Compatible drive types • 2 x 3.5" or 2.5" SATA HDD/SSD (drives not included.)
- External ports 2 x USB 3.2 Gen 1 ports
- Form factor Desktop Size (H x W x D) 165 x 100 x 225.5 mm
- Weight 0.88 kg
- LAN ports 1 x 1GbE RJ-45 Wake on LAN/WAN Yes

- Scheduled power on/off Yes
- System fans 1 x (92 x 92 x 25 mm)
- AC input power voltage 100 V to 240 V AC
- Power frequency 50/60 Hz, single phase
- Operating environment
    - Temperature: 0°C to 40°C (32°F to 104°F)
    - Relative humidity: 8% to 80% RH
- Storage environment
    - Temperature: -20°C to 60°C (-5°F to 140°F)
    - Relative humidity: 5% to 95% RH
- Maximum operating altitude 5,000 m (16,400 ft)

Another example of an embedded system is the system enclosed in the AC unit. It contains the actual cooling unit (indoor unit, outdoor unit which contain evaporator, condenser, expansion valve and compressor), a remote control (for the user) and a remote-control receiver, command interpreter (software) and a controller unit.

The basic principle of operation relies on the fact that compressing a gas produces heat, which can be removed, and then when expanding the same gas, it will go back to a temperature lower than the initial temperature. This can be used to cool the air in a room.

The control unit has a closed loop which continuously monitors the temperature in the room and the desired temperature set by the user.

Assuming it is summer, and the user wants a lower temperature than outside, if the temperature in the room is higher, it will start or increase the speed of the compressor, to cool down more. If the temperature in the room is lower, it will stop the compressor and wait.

Another example of an embedded system is a system made to assure water to some plants. The system will be composed of: humidity sensor to convert humidity in voltage, water pump, and a microcontroller having a program that periodically check the humidity sensor using an ADC (analog to digital converter), compares it to a preset value, and it the soil is too dry, sends command to the water pump for a certain amount of time, to increase the soil's moisture. Notice the green and red lines below with assure the feedback of the control loop.

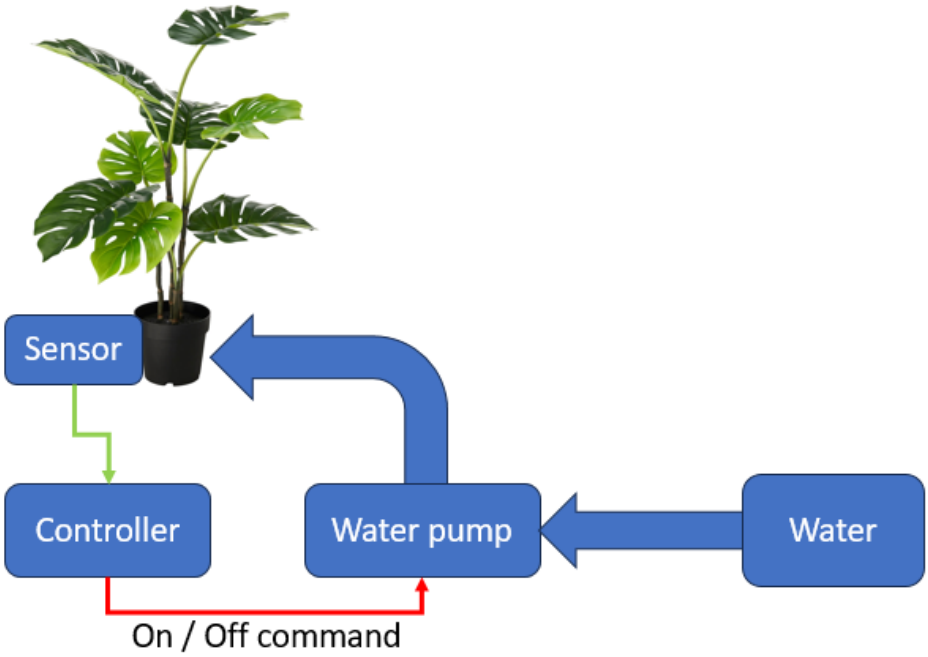


Figure 137. Automatic plant watering system

5.2.2 High-performance clusters

Clusters allow packing a lot of compute power and storage in a small footprint, but it usually active cooled to ensure longevity. Currently (2023) the top world supercomputer is Frontier, a cluster of 22700 MW, 700 Petabytes for storage, 8.7 million computing cores (AMD EPYC 3rd generation) running at 2 GHz grouped in 9400 nodes, and connected by 90 miles of networking cables.

Figure 138. A cluster organized in cooled racks

5.2.3 Personal Computers / Small servers

Personal Computers have often been utilized for various projects where more processing power is required than a small embedded system. When reliability is very important, they were rugged for vibrations, they were weather proofed or transformed (using redundant power supplies and forced cooling) into more reliably processing systems. The advantage of using off-the-shelf components is the price and availability.

Most common case for servers, which resembles a desktop or workstation yet still may be mounter in a server rack, is 4U format. 4U refers to the height of the used slots in the rack. Racks may be multiple widths, but most common is the 19".

Figure 139. Examples of various-sized 19"racks

The smaller racks (which are often wall mounted) are usually used for storing network components, whereas the larger ones (taller) are used for servers. Although 1U and 2U cases may increase performance density, they also increase heat so forced cooled is usually present in such high-density computing devices. 1U equals to a height of 44.45 mm (1.1750") and is the smallest height that may be mounted safely in a rack.

4U cases have the advantage of being able to accustom large amounts of RAM memory (up to 64 slots), 2 or 4 CPU, large amount of SSD drives, and a few GPUs at a price which is close-enough to the regular personal computers. Server motherboards may offer up to 8 PCI-express slots and depending on the CPUs utilized, 100+ PCI-express lanes.

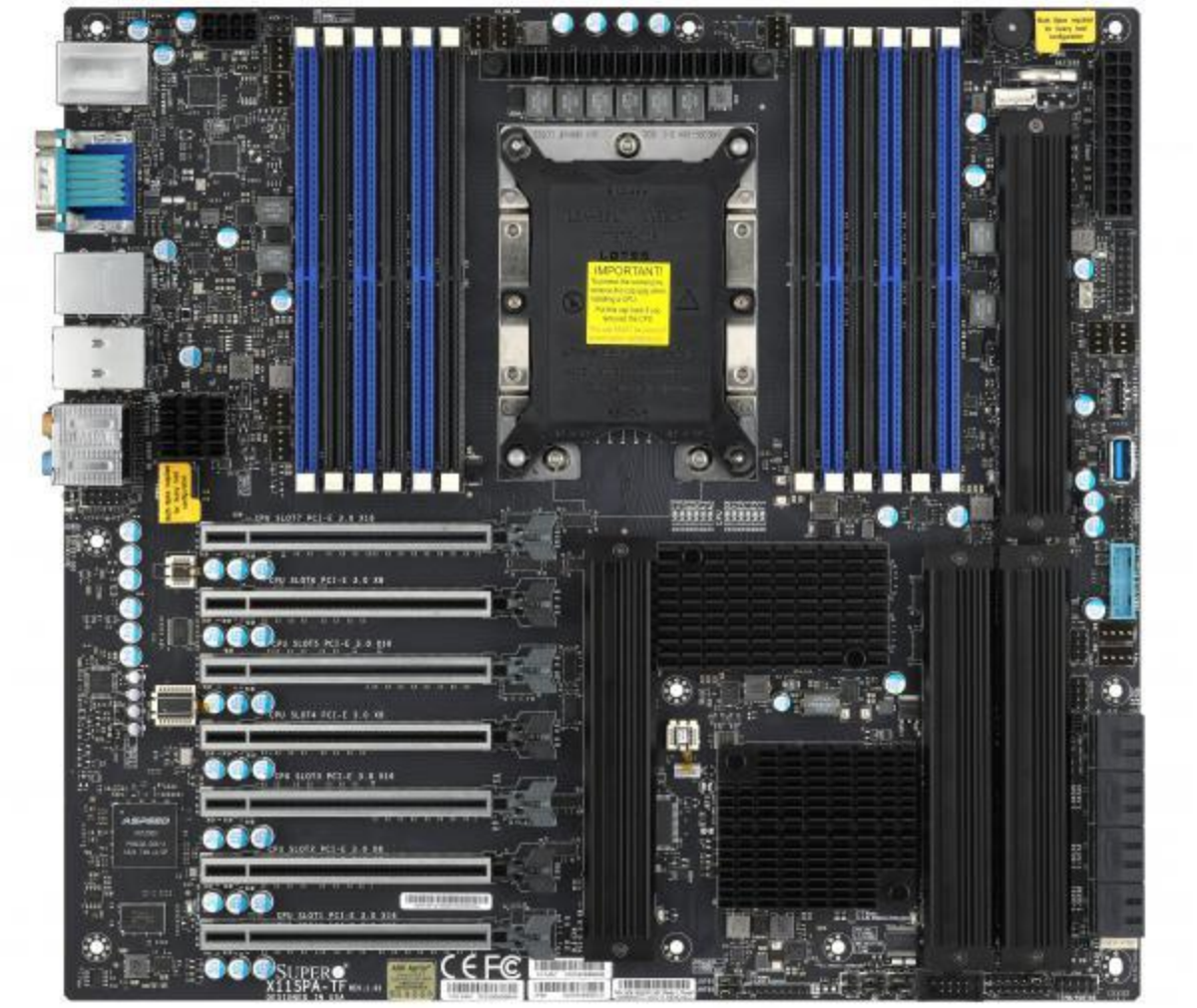


Figure 140 The Supermicro X11SPA-T motherboard

The above motherboard may hold up to:

- 3 TB of RAM, in 12 memory slots
- 7 PCI-e slots: 4 of x16 and 3 of x8
- M.2 interfaces: 4 supporting RAID 0,1,5 and 10
- 3 network interfaces (one is for IPMI, one of 1 Gbps and one of 10 Gbps)
- 2nd gen Scalable Xeon processor with up to 28 cores and TDP of 205W
- 6x USB3.1 Gen1, 4x USB3.1 Gen2, 2 USB2.0
- 2x serial ports
- Voltage sensors on VBAT, memory, HT, 3v3, 5v, 12V, 1V8, fan status
- 10x 4-pin fan headers (with rpm measurement and control)

5.3 NVM cycling application design

For more details regarding this application, refer to paper [29], which is an extensive work description on this subject. The following text is a walkthrough for the process of designing, implementing, and testing a real mixed-signal application, the NVM cycling application.

5.3.1 Problem description

Natural language: The flash memory has a limited number of erase-program cycles. Therefore, flash memory manufacturers need to test their prototypes before large-scale manufacturing. A test should be devised, to run as fast as possible on many devices.

5.3.2. State of the art

Except for one time, all the other times when this problem occurs, somebody else faced the same problem before (with some modifications) so it makes sense to check the state of the art (previous work)

5.3.2 Specifications

Specifications must be extracted/defined such that they may be a pillar to lean on, when the wind of research blows over oneself.

The test setup should meet the following items:

- allow testing of high number of samples/chips (tens to hundreds)
- temperature typical 25˚C, but should be possible from -40˚C to 105˚C (in compact thermal chamber)
- all wear leveling mechanisms disabled by software.
- in case of an error, it must be easy to determine where (sector, page) and when (cycling counter) it occurred.
- ability to stop and resume later from the same point.
- allow variable external voltage and frequency.
- compact size
- easy to manufacture (commercially available parts)
- test speed must be limited by chip and not by the setup.

Some requirements regard only the hardware, some regard only the software and some must be implemented in both.

5.3.3 Estimations

Always do estimations: they will help the design process, and make very clear what is achievable, even before one single line of code being written. This will save a lot of time

(instead of exploring dead-ends). In computer algorithms like backtracking, this is called pruning and can heavily decrease the time and space complexity.

Since an 128 Bytes of flash memory needs 2.3 milliseconds to be erased or written, on a chip having 512 KB of flash memory, one would expect: 2*2.3ms*512 KB / 128 B = 18.84 seconds to run.
Considering a target of 100k loops (for endurance test), one would expect 21.8 days of runtime.
Since the test will happen on at least 100 samples, a one-to-many communication scheme must be devised (we cannot have one reader per chip) => the model will be one master – multiple slaves, and the master will continuously poll each slave for status.

5.3.4. Pseudocode for host

The host application will run on a PC, which has a smartcard reader connected to it.

```
function PC_Main() {
        - connect to MUX_CTRL or die with error
        - connect to ISO reader or die with error
        - get ATR from all DUTs
        switch (USER_WISH) {
        case DO_DOWNLOAD_LOG:
                - print info about current test (temperature, voltage, frequency,
                        number of erase/program cycles)
                -sweep all DUTs and get their logs with failures
        case DO_RESUME_TEST:
        case DO_START_TEST:
                - if (DUT application version is NOT latest) then download latest
                - erase/program app on DUT
                - print info about current test (temperature, voltage, frequency,
                        number of erase/program cycles)
                - sweep all DUTs and send command to continue or start the erase/program test
                - while not all DUTs finished test {
                        every 10 minutes:
                        - sweep all DUTs and check their status
                        - print matrix with DUTs' status
                        - if all DUTs finished except a few that hung, wait for a
                                "guard time = enough time to finish test" and inform
                                the user about this waiting
                }
                default: signal error ("not a valid user wish")
        }
        print matrix with DUTs' status
        disconnect ISO reader
        disconnect MUX_CTRL
}
```

Figure 141. Host-side pseudocode

5.3.5. Hardware design

The design of the hardware part will be governed by the physical dimensions of the full setup. The thermal chamber will set the physical shape of the setup, otherwise it would make sense to build a cubic-shaped setup to minimize distances and keep it portable.

Figure 142. Votsch temperature test chamber

If the chips come in CDIP24 format, a layout of 5 boards each having two rows of 10 chips each would prove advantageous. This will keep the area footprint small enough to fit in the thermal chamber, yet the height is small enough to be easy to handle. Since the chips will be changed often, the slots should be of ZIF (Zero Insertion Force) to avoid damaging the slots.

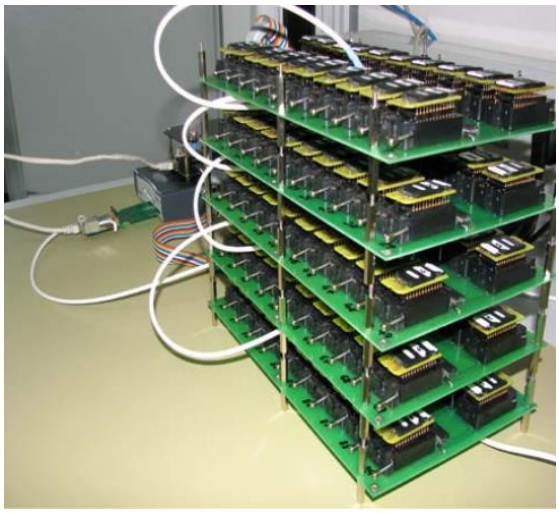


Figure 143. The finished hardware test setup

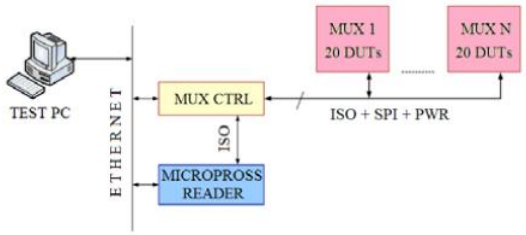


Figure 144. Block schematic of the system

The MUX boards is a board that can provide power to 20 chips simultaneously, and multiplex all data lines towards the master (the single reader). It uses analog switches (ADG719) and an IO expander (MCP23017/MCP23S17) to do this. The analog multiplexers were chosen to support the variable voltage required by specifications (1.8V … 5V), similarly, the IO expander can support the voltage and the data lines at those voltages (if it would not, we would use level translators).
The MUX_CTRL board is a microcontroller board supporting Ethernet and RS232 interfaces and is utilized to command the MUX boards via SPI and IO expander, which one chip to be linked with the master (reader). All the chips are powered continuously, but only one chip may exchange data with the master at a time.

5.3.6. Software design

The software written will run on the following systems:

- The DUT (device under test) software, which takes commands from the master to start cycling, stop cycling, resume cycling and retrieve status (waiting, error, running)
- The PC (master) software which will govern the whole operation, select them serially, ask the chips to identify themselves and start/stop/pause/resume cycling. It will also create logs (so that a loss of power is not a loss of time) and updates them often (every 10 minutes) during the expected 22 days of running.
- The MUX_CTRL which is a microcontroller, interpreting the commands from master, and controlling the IO expanders and analog multiplexers such that selection of the chip can be made. It also relays signals (data, reset, clock of the ISO7816 interface) to the selected chip (DUT)

5.3.7 Outcome

Using commercially available components a highly reliable setup for testing flash memory Endurance was built. It has already been successfully used for the evaluation of more than ten distinct microcontroller types. This approach of testing the flash endurance of microcontrollers is unconventional. Still, it allows the optimal implementation of the test setup with the following major benefits:

- the setup supports the testing of up to 100 DUTs simultaneously (but can be extended further by adding more MUX-devices)
- test run time is dictated only by the DUT (more precisely, the flash write/erase times), which is great, since faster chips will run the test faster

- even though industrial grade components were used for the implementation, the price of the hardware was around 1500 EUR

5.4 Digital signal processor for RADAR systems

For more details regarding this application, refer to paper [30] and [33], which is an extensive work description on this subject. The following text is a walkthrough for the process of designing, implementing, testing of a real mixed-signal application, the ground-based digital signal processor of a LEO RADAR. LEO (Low Earth Orbit) is the orbit with a period of 128 minutes or less and eccentricity less than 0.25

Figure 145. Common orbits around Earth

5.4.1 Problem description

Natural language: The RADAR emits electromagnetic waves (continuous or in pulses). When using continuous wave (requires lower emission power) one must use two antennas to avoid reducing the dynamic range: one for TX and one for RX. The TX signal is built amplified and emitted by specialized equipment, whereas the RX signal is receive by the analog front-end, amplified and sampled to be analyzed for the purpose of obtaining the range (distance from RADAR to object) and radial speed (relative speed towards/away from RADAR). The RX signal is the signal reflected by the target.

Unmodulated, continuous wave RADARs cannot detect static objects or measure the range to an object/target. Therefore, one should use a type of modulation (usually frequency modulation, therefore FMCW) to send up-chirp (signal of increasing frequency) and down-chirp (signal of decreasing frequency) such that, the measured difference between the Tx

signal and Rx signal (called beat frequency) which is a function of range and speed of light., can be used to compute the range. Linear modulation is often used.

In addition, if the object is moving, one can use the Doppler effect to compute its velocity towards the RADAR. This will alter the beat frequency with a factor depending on this velocity (called radial speed). Two measurements should be performed to get two equations where Range and Velocity are the unknowns.

$$f_{beat} = (\Delta F / Tm) * (2R / c) + 2 v_r / \lambda$$

Where:

- f_{beat} is the beat frequency
- ΔF is the total peak-to-peak frequency deviation (modulation bandwidth)
- Tm is the modulation period
- R is the range to the target (m)
- c is the speed of light through void (m/s)
- v_r is the radial velocity (m/s)
- λ is the wavelength

2R/c is the echo time, the time it takes for the signal to go from TX antenna to the object and back into the RX antenna. For more details regarding the mathematics of this, one may address [31]

Figure 146. Cheia1 RADAR's antenna (32 meters wide, 309 tons of weight)

The picture in Figure 139 was taken at Cheia site (Romania, 100 km away of Bucharest), and we may see, one of the two 32 meter-antennas, made under Japanese license in 1977-1979. The signal processor was part of the Cheia Retrofit project [32].

5.4.2. State of the art

Except for one time (first one), all the other times when this problem occurs, somebody else faced the same problem before (with some modifications) so it makes sense to check the state of the art (previous work). The state of the art revealed the following data:

Name[paper]	ADC	Signal processor	Peak Power	Comments
BIRALET[35155]	2x 14 bit @ 10 MSa/s	USRP 2954R w Matlab	10 kW	two antennas 20 km apart, 7m antennas P-band (420 MHz)
ATLAS[36]	2x 16 bit @ 100 MSa/s	Advantech MIO 2360	5 kW	Monostatic pulse radar, 5.56 GHz
S3TSR[37]				Close monostatic L-band (1400 MHz), electronic scanning arrays for TX and RX antennas
TIRA[38]			2 MW	1.33 GHz
GESTRA[39][40]			256 kW	L-band (1280-1380 MHz), 256 transmitting elements
GRAVES[41]				VHF (143.05 MHz)
CAMRa[42][43]	7x 12 bit @ 2 MSa/s	Pentium PC with DSP cards	600 kW	25m antenna, S-band (3076.5 MHz)
ACROBAT[42]	3x 12-bit		2 kW	25m antenna, 1275 MHz

One thing to note is that people are not usually very specific with details, especially in military space radars.

5.4.3 Specifications

A system schematic was available and the role of the signal processor (SP) was stated by the system designer. The TX antenna, Rx antenna, power amplifier, signal generator, master oscillator, storage server and monitoring and control server, were built by other companies.

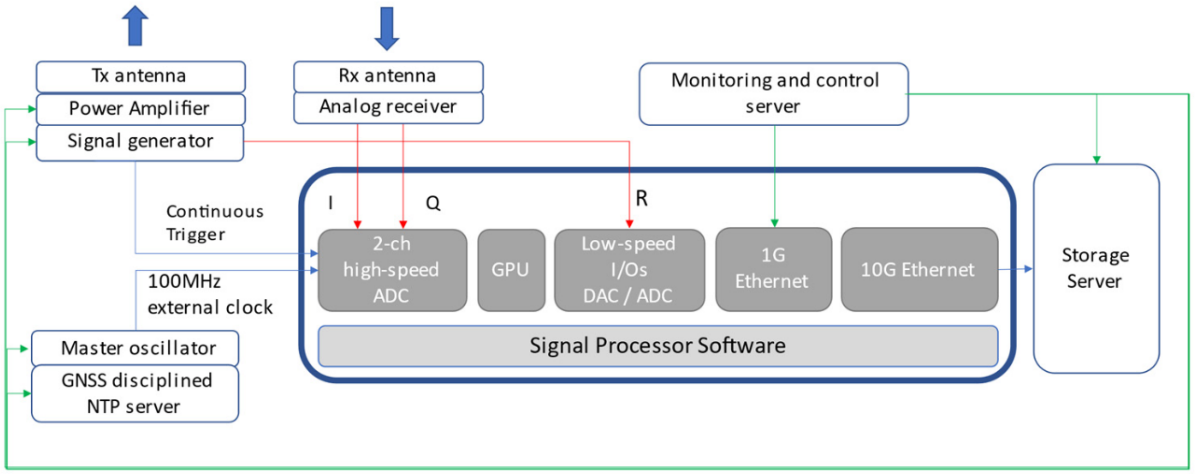


Figure 147. System schematic of the RADAR

Figure 140 shows a system schematic of the radar. In the center—the signal processor (blue rectangle). The antennas transmit the signal and receive the echo from the space object, the master oscillator provides a very reliable, very precise external clock for the SP's ADCs, the monitoring and control (M&C) server offers the user interface and manages the system, and the storage server stores the acquired data. Red arrows denote analogue signals, green arrows denote digital signals [30]

Performance specifications

The SP should ensure real-time data processing of incoming data or at least near real-time, but with no data lost. Analog data are sampled at 100 MSamples/s using 4 channels (two ADCs with two channels each) of 16-bit; this results in a data rate of 100 M × 4 × 16 = 6.4 Gbps

As seen in Table 28, multiple scales for distances between 400 and 4000 km were used (and accordingly multiple settings for RF modulation and digital processing); this is due to the following reasons, according to the system designer [155]:

- It allows to keep a constant, relatively small RF bandwidth (2 MHz) which allows the receiver chain to keep noise under control (the noise is a product of bandwidth with spectral density). Constant bandwidth allows simpler receiver design, and one would prefer to modify the parameters of the receiving pipeline, in software (digital domain) in the SP, as opposed to temper with the analog part in the receiver. The complex receiver means complex tuning across different frequencies (we use frequency modulation).

- the power scattered by the target (the echo signal) decreases with a power of 4 of distance, so the further away the target is, the much worse would be for the SNR. This requires a much better noise cancellation algorithm, which requires a larger FFT, which requires a longer signal in time (continuous emission over larger distances requires a longer time-duration of the signal due to longer round-trip time).

There is only one main performance requirement, and it is for the processing time of an acquired waves.

Table 28. Planned (by design) scales for the RADAR

Scale Number	Description (Range)	Range (km)	Trigger Period (ms)	Acquired and Decimated Wave size (samples)*	Acquisition Time (ms)	Maximum Processing Time (ms)
1	Close	< 400	40	2x 256 k	13.1	40
2	Mid-close	< 900	100	2x 512 k	26.2	90
3	Mid-far	< 1800	250	2x 1 M	52.4	180
4	Far	< 3900	500	2x 2 M	104.8	390

*2x comes from: one wave for UP chirp and one wave for DOWN chirp

From the last line of table, one may see the maximum amount of data to be processed is 2x 2M samples on 4 ADC channels, coming every 390 milliseconds. The acquisition was later changed from 2M every 290ms to 7M (2M for UP, gap, + 2M for DOWN), due to the ADCs' lack of hardware timestamping.

The estimated acquisition time for scale number 1 is 256k Sa @ 100 MSa/s / 5 (for decimation with factor of 5) => 13 milliseconds, whereas for scale number 4 is about 105 ms. Using multiple ADCs in parallel, will parallelize data acquisition, but not the data transfers.

5.4.4 Design /choice of the hardware

Hardware was constrained to use high speed / high precision ADCs, and one of the top choices was the AlazarTech ATS9462 [155], 16-bit, 2-channel PCI express card.

The initially expected data processing was to be performed in a:

- FPGA (harder to design, but reliable), in a SoC setup with ARM + FPGA in the same package
- GPU (easier to code, but need special software knowledge)
- Multi-core CPU (easiest to code)

Literature review in FPGA design for FFT (one of the expected operation) stopped at 1-2 Million samples, so it deemed less flexible (and kept as backup); in addition, some extra data processing was expected to be required and implemented everything in a FPGA or on a lower performance CPU was risky and time consuming. For the second and third choice, it implied using a server or workstation as a host for the signal processor. The preferred choice was made for a 4U server, since components (CPU, cooling, redundant power supplies, IPMI-dedicated NIC for BIOS-level access) are more suited for high up-time / reliable operation and remote administration. This was a great choice since it allowed multiple lanes of PCIe connected in the system (most Intel i3/i5/i7/i9 have only 16 lanes, AMD Ryzen 7 have 20,

whereas current Intel Xeon / AMD Threadripper support at least 48/64); GPU card alone will take 16 lanes.

According to the FSM of the signal processor, the main operations that should be implemented are:

- Data acquisition using the high-speed ADCs
- Data (error signal) generation via PCIe DACs
- Data transfers from inputs to CPU/GPU-memories, between software components, and towards the outputs
- Data processing:
    - Decimation which is of O(N) complexity
    - Data conversion (from 16-bit integers to fp32 or fp64), O(N) complexity
    - Complex-to-complex 1D FFT which is of O(NlogN) complexity
    - ABS (computing absolute value of vector with complex numbers), which is of O(N) complexity
    - CFAR which is of O(N) complexity

Checking the list above, revealed two constraints:

- Space constrains: RAM size enough to store one triggered acquisition (if not enough speed to process it in real-time)
- Speed constrains: the FFT (highest complexity) should be performed (much) faster than the available time

Next step, was to count the PCIe cards to be hosted:

- 4x AlazarTech ATS9462 ADCs cards
- 1x GPU (multi-slot, to speed up computation if needed)
- 1x slow DAC card (to output error signal)
- 1x slow ADC card (to capture trigger timestamp)

A motherboard with at least 7 slots was selecting, capable of hosting a Xeon CPU with multiple threads.
Regarding the choice of more CPU threads vs higher CPU speed, we estimated the amount of software threads:

- High-speed data acquisition, 1 thread, throughput: 4 ADC cards, 2-channels, 16-bit each, 100 MSa/s => 4*2*(16/8)*100 M/s = 1.6 GB/s
- Low-speed data acquisition, of timestamp: 1 kSa/s, 1 bit, for timestamping => low workload
- Low-speed error signal output, once every 40…390 ms => low workload
- High-speed data processing: decimation, FFT, abs, CFAR => high workload
- Command-interpretation (via Ethernet, using TCP/IP server), 1 or 2 clients, a few messages per second => low workload

Since the two main high-demand software threads were data dependent, the choice was made for a CPU with at least 3 cores, but of higher frequencies => Xeon W-3223, 8

cores/16T running at nominal 3.5 GHz (instead for example, Xeon W-3275 with 28 cores running at nominal 2.5 GHz).

Next step was benchmarking the CPU to check for necessary computing power. The first one to be benchmarked is the FFT, then all the others will follow as a complete performance profile should be extracted, for this application. Benchmarking was done at multiple sizes, to see exactly how it scales and whether it worths bringing in, a GPU or a multi-core CPU is enough. All data in Table 29 was extracted while running in OpenCL and averaging 1000 loops to reduce measurement errors.

Table 29. FFT measurements in CPU and GPU setups

FFT size	Machine1 (using GPU)		Machine 2 (using CPU)		Machine 2 (using GPU)	
[samples]	[milliseconds]	[cycles]	[milliseconds]	[cycles]	[milliseconds]	[cycles]
64	0.1	116270	1.5	385000	0.1	112050
256	0.1	52850	0.4	175000	0.1	112050
1K	0.1	84560	0.8	280000	0.1	124500
4K	0.1	95130	0.3	315000	0.2	236550
128K	0.5	496790	1.5	2E+06	0.2	273900
256K	0.8	803320	2.4	3E+06	0.3	361050
512K	1.6	2E+06	3.8	6E+06	0.4	547800
1M	3.5	4E+06	7	1E+07	0.9	1E+06
2M	7.6	8E+06	13	3E+07	1.8	2E+06
4M	17	2E+07	24	6E+07	4.1	5E+06
8M	39	4E+07	85	1E+08	7.3	9E+06

Machine1: Laptop Asus ROG Flow X13 having:

- CPU: Ryzen 9 5900HS 8C/16T @ 3.0 GHz
- GPU: NVIDIA GeForce RTX3050 @ 1057 MHz
- RAM: 32 GB DDR5 @ 4266 MHz

Machine2: 4U Supermicro Server with X11SPA-T motherboard having:

- CPU Xeon W-3223 8C/16T @ 3.5 GHz
- GPU: NVIDIA Tesla V100s @ 1245 MHz
- RAM: 320 GB DDR4 @ 2666 MHz

From the above table one may conclude that staying on the CPU, might be possible: we have 390 ms to process two blocks of 2M data with operations that include FFT. For FFT alone it should not take more than 26ms. The other operations were benchmarked further into the project and the Table 30 was obtained.

- DEC is decimation by averaging
- ABS is complex absolute value of vector values
- CFAR is Constant False Alarm Rate applied with Cell Averaging (CA)

CFAR operation sets a threshold depending on the neighboring frequencies, and then applying that threshold retrieves the peaks which are the expected beat frequencies. Depending on how the threshold is computed it may be CA (left + right / 2), GO (max (left, right)) or LO (min(left, right)), where left and right are the averaged value of left/right neighboring cells. Usually a small number of cells is skipped between the cell under test and left/right neighbors (these are called guard cells)

Table 30. Benchmarking all operations for 2M samples in CPU and GPU setups

Operations (ops)	GPUa (ms)	CPU (ms)
DEC	60	16
COPY	15	1
FFT	15	5
ABS	15	5
CFAR	16	13
WIN	15	28
FFT_ABS	16	40
FFT_ABS_CFAR	17	25
WIN_FFT_ABS_CFAR	17	30

To process data, one has to bring it to local memory, then run the processing algorithm, and send the result back. This proves to be a waste of time when using a GPU, as it doesn't really make sense to send the data from the GPU memory to CPU memory, and back, for the next GPU operation. Therefore, chained operations were implemented (intermediary results stay in GPU memory and GPU kernels run on them), thus reducing the total windowing, FFT, absolute value and applying CFAR, from a sum of 61ms to 17ms, per block of 2M. Since two blocks must be computed for every trigger, we reduced the time from 122 ms to 34 ms.

The three integrated NICs were used for:

- High speed data transfer of debug data, to storage server (the 10 Gbps one) – measured transfer rate of 392 MB/s to 525 MB/s
- SP's server configuration and command (the 1 Gbps one) measured average turnaround time of under 9 milliseconds.
- Remote management (the IPMI one)

5.4.4 Design of the main software

The software was built to mirror the block schematic of the signal processor's FSM. All software parts that can run in parallel were implemented as threads.

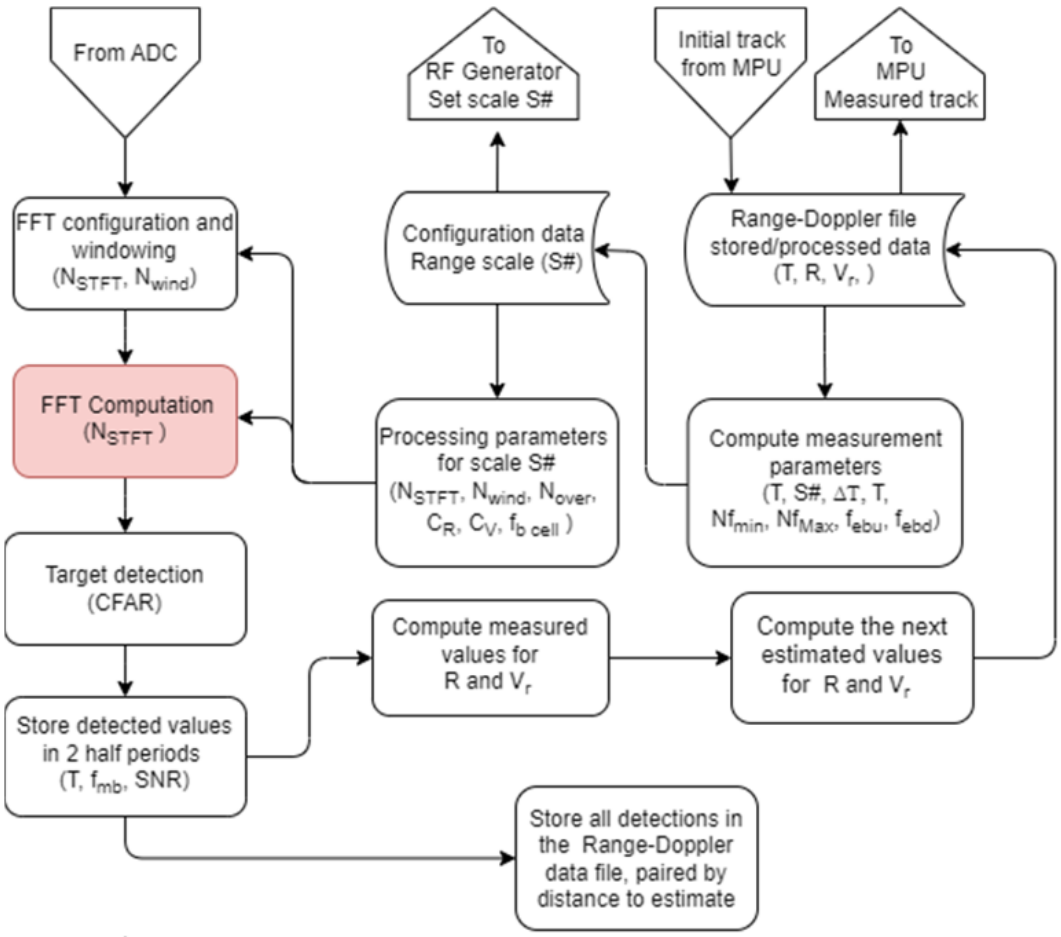


Figure 148. The FSM of the signal processor

Figure 148 presents the finite state machine of the signal processor. Digital data flows inside from the ADCs, and they are processed by decimation and windowing, FFT, absolute value, and CFAR. The FFT processing is highlighted in red, as this has a very large size, quite unusual in a radar system.

5.4.5 Implementation and test

All software blocks were written and tested against Octave scripts (with accurate data processing implemented in public libraries by others). A simulator, based on previous runs, was developed, to be able to run tests, even if no satellites are in range.

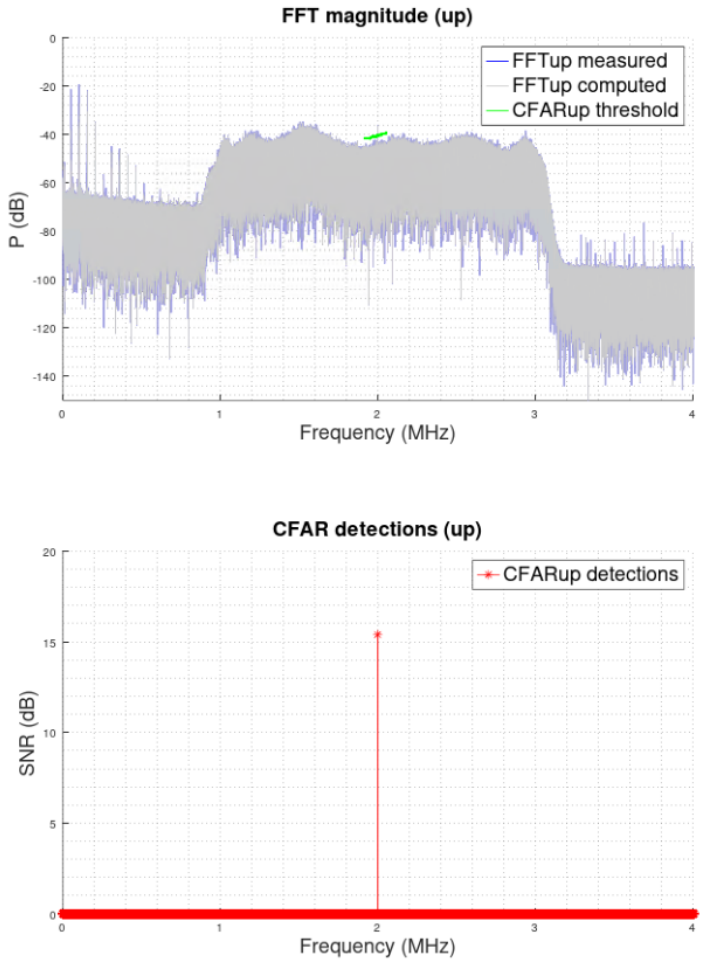


Figure 149. Data processed in SP compared against Octave (top). Peak detection after CFAR (bottom)

A remote management interface was implemented in Java to allow manual configuration, control, testing and debug of the signal processor.

5.4.6 Conclusions

The software processing implementation worked for the estimated part, however, data transfers were previously overlooked and proved to be quite time-consuming. For example since the high-speed ADCs are PCIe x4 Gen1, the maximum data rate they are capable of, is 250 MB/s: for scale 4 for example data transfers will take around 2x 2M * 16-bit / 250 MB/s = 32 ms. In addition, the PCIexpress slot where the GPU was hosted, was downgraded by the motherboard to x8, slowing down data transfers (a consequence of having 2x NVMe memory, with PCIe x4 interface). All in all the deadlines were met, with some spare in case of scale 3 and 4, but had to increase the available RAM memory, to keep unprocessed data for scale 1 and 2, when complex data presented multiple peaks after the CFAR operation.

5.5 E-Health embedded system for choice reaction time (CRT) measurement

An extended description and motivation for this system may be found in [44] and [45]. In natural language, the human subject needs to press as fast as possible, right button or left button depending on the color an RGB led shows. Data should be uploaded into cloud at the end of the experiment with some statistics attached. The purpose is to make the human subject choose between a target color and two distractions and measure the time it takes to choose.

Specifications

The setup should:

- be portable, and lightweight
- contain an RGB LED
- contain two tactile switches
- contain a battery capable of powering the setup for at least 30 experiments
- contain a battery charger, powered by USB connection
- save logs on correct/incorrect button presses
- upload logs using Wi-Fi, into an Internet-connected server.
- have the probability of occurrence per color as close to 33% as possible, and the occurrences must be random (not predictable, from experiment to experiment)

The target color is BLUE and the correct button for it is RIGHT button. The distraction is the GREEN and RED colors, and the correct button for them is the LEFT button.

Software design

The FSM of the software should implement the schematic in Figure 150: a full experiment takes 100...300 seconds, that is, a maximum of 40...120 rounds with minimum SOA (stimuli onset asynchronies) time of 2.5 seconds each. The LED is lit for 200ms and then shut off. The human subject is expected to press a button to confirm the color. If the correct button was pressed, the device will allow a SOA time to pass, then it will go to the next round. If the incorrect button was pressed, the device will wait until the correct button was pressed, then wait for the SOA time to pass, and progress to the next round. The experiment ends when the experiment time elapsed. The log with statistics (how many button presses were correct, how many incorrect and how long it took to get the response, per LED color) is sent into the cloud at the end of the experiment. Initial implementation utilized interrupt driven software, but after problems with software updates (bootloader was not working properly when chip was in sleep), another implementation was done, using cooperative multitasking. This

allowed, similarly as with interrupts, to monitor other sensors as well, while measuring the CRT time. All hardware modules have their counterparts in C++ classes.

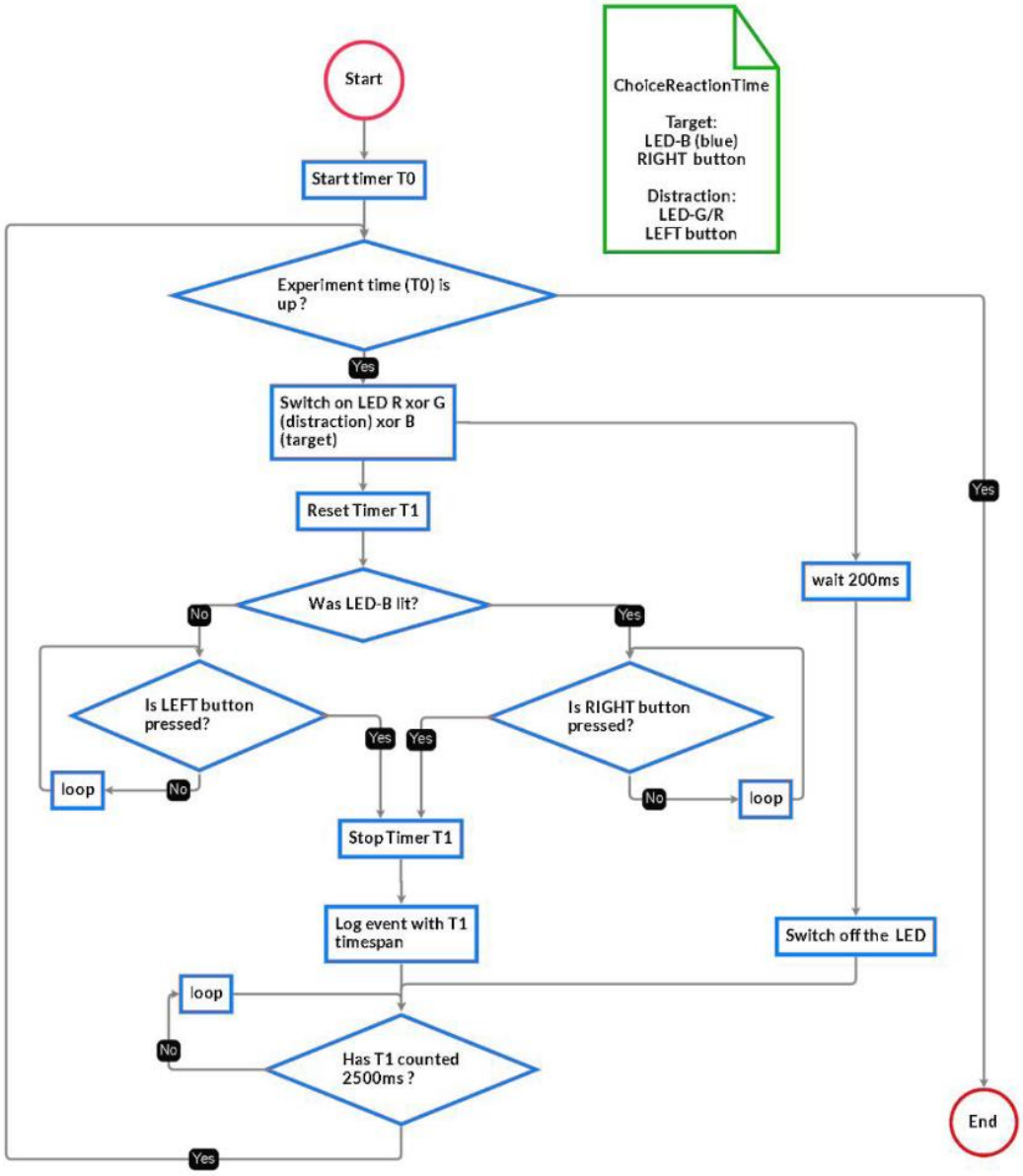


Figure 150. FSM of the CRT system (without data sending)

Hardware design

A MCU board, with Wi-Fi and I/O edge-triggered interrupts, strong output drivers (capable of driving directly the LEDs) and C++ compiler support. The RGB LED was chosen to be common anode as the MCUs have usually greater sink capability. The LED current was limited with 1kOhm resistor per color terminal, as shown in Figure 151:

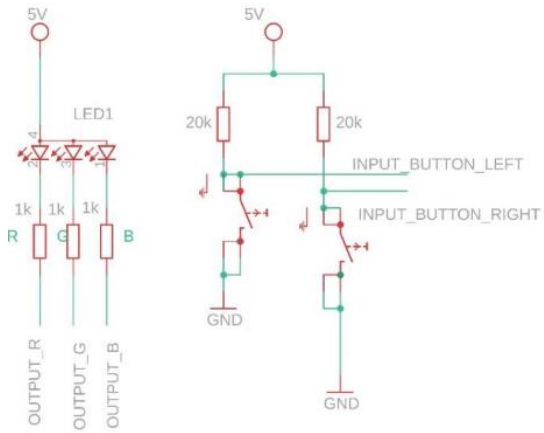


Figure 151. CRT equivalent electrical schematic

Depending on the MCU board one may power the LED at 3V instead of 5V (but the resistors should be decreased, to keep the luminosity the same). Average consumption was measured at around 33 mA (3mA come from the LED) but can be reduced by sending the chip into sleep. Assuming average experiment size of 300 seconds, the necessary energy consumed for it, is (33mA * 300s) / 3600s = 2.75 mAh. The specification required a battery life of at least 30 experiments, therefore the battery should be chose to have a capacity of at least 83 mAh. A choice was made to add a power button (to keep consumption at 0 mA, when not used), and the battery was chosen by the physical size and weight (for example a 700 mAh only measures 33x33x6mm and weights 15 grams).

Testing

The system was tested in 5-minutes long experiments, and the results obtained were as seen in Figure 152. An analysis of the Red/Green/Blue probability of occurrence gave results between 32.22 % / 34.73 and respectively 34.73%

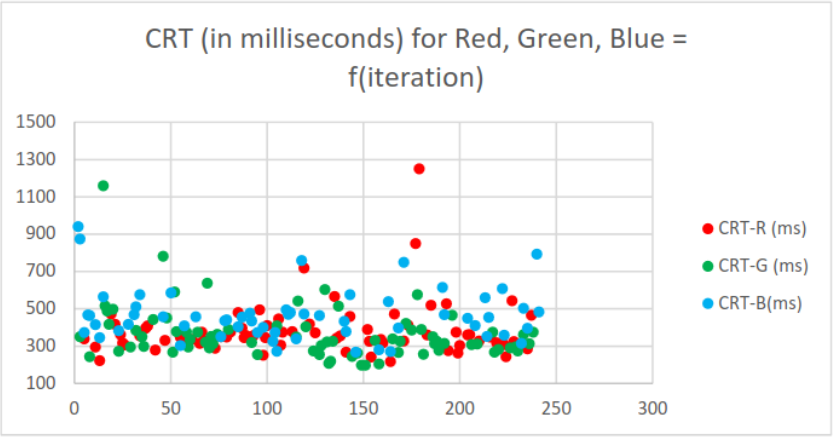


Figure 152. CRT (ms) for each color as function of iteration

6. Conclusions

This book provided a presentation of the mixed-signal system design topic. It started with a talk on the digital and analog signals, then progressed to data processing and data communication, then it introduced the basic bricks a system designer uses to create a system (both software and hardware), and finally, it presented a few examples of embedded, embedded + PC, and PC-based systems with the rationale and details behind the concept, design, and implementation of such real-life systems.

Figure References

1. Figure 1. A digital signal (as a result of both sampling and quantization processes). Digital Integrated Circuits https://wiki.dcae.pub.ro/index.php/Introducere._Verilog_HDL
2. Figure 2. A time-sampled signal (as a result of sampling process). Digital Integrated Circuits https://wiki.dcae.pub.ro/index.php/Introducere._Verilog_HDL
3. Figure 3. A value-sampled signal (as a result of quantization process). Digital Integrated Circuits https://wiki.dcae.pub.ro/index.php/Introducere._Verilog_HDL
4. Figure 4. AMD Athlon64 x2 CPU (left is top, right is bottom) with Socket 939. AMD Athlon 64 X2 6400+ Black Edition Review, https://www.trustedreviews.com/reviews/amd-athlon-64-x2-6400-black-edition
5. Figure 5. 32GB DDR5-4800 RAM server memory with error correction code (ECC). Micron 32GB DDR5-4800 ECC UDIMM 2Rx8 CL40, https://eu.crucial.com/memory/server-ddr5/mtc20c2085s1ec48ba1r
6. Figure 6. RTX4090 GPU card with PCIexpress x16 slot, without passive and active cooling installed. https://techgage.com/article/nvidia-geforce-rtx-4090-the-new-rendering-champion/
7. Figure 7. A SATA-interfaced SSD (Solid State Disk) for non-volatile storage. 860 EVO SATA 2.5" SSD 250GB, https://www.samsung.com/us/computing/memory-storage/solid-state-drives/ssd-860-evo-2-5--sata-iii-250gb-mz-76e250b-am/
8. Figure 8. A PCIe-interfaced SSD for long term storage. 980 PRO PCIe 4.0 NVMe™ M.2 SSD. https://www.samsung.com/nl/memory-storage/nvme-ssd/980-pro-pcle-4-0-nvme-m-2-ssd-500gb-mz-v8p500bw/
9. Figure 9. A comparison between SLC, MLC, TLC and QLC cells https://www.kingston.com/en/blog/pc-performance/difference-between-slc-mlc-tlc-3d-nand
10. Figure 10. A monitor with common pixel resolutions (HD = High definition).The right monitor for yout home office – EIZO. https://www.eizo.co.uk/knowledge/workstation-ergonomics/the-right-monitor-for-your-home-office/
11. Figure 11. RGB (red, green, blue) pixels in monitor image that spatially combine to obtain all colors (white color is all RGB leds in that pixel lit). TV Pixels – RTING.com. https://www.rtings.com/tv/tests/picture-quality/pixels
12. Figure 12. RAID0: high speed, no redundancy https://www.seagate.com/gb/en/manuals/network-storage/business-storage-nas-os/raid-modes/
13. Figure 13. RAID1 mode: data mirroring https://www.seagate.com/gb/en/manuals/network-storage/business-storage-nas-os/raid-modes/
14. Figure 14. RAID5 mode: single drive redundancy https://www.seagate.com/gb/en/manuals/network-storage/business-storage-nas-os/raid-modes/
15. Figure 15. RAID6 mode: dual drive redundancy https://www.seagate.com/gb/en/manuals/network-storage/business-storage-nas-os/raid-modes/
16. Figure 16. RAID10: a RAID 0 of RAID1s https://www.seagate.com/gb/en/manuals/network-storage/business-storage-nas-os/raid-modes/
17. Figure 17. Anandtech's comparison of Intel P4510 SSD (x4 PCIe) drives in various RAID configurationshttps://www.anandtech.com/show/12435/the-intel-ssd-dc-p4510-ssd-review-part-1-virtual-raid-on-cpu-vroc-scalability/4

18. Figure 18. OSI layer model for networks. An OSI Model for Cloud – Cisco Blogs. https://blogs.cisco.com/cloud/an-osi-model-for-cloud
19. Figure 19. TCP vs UDP in how data is transmitted an the possible types of connections. TCP vs UDP – How Are They Different? https://cheapsslsecurity.com/blog/tcp-vs-udp-how-are-they-different/
20. Figure 20. Visual examples of network topologies. Types of Netwok Topology – TechieReader, https://techiereader.com/types-network-topology/
21. Figure 21. Resolving a website's server address by using DNS hierarchy. What Is A DNS Server? How Domain Name System Servers Work. https://www.businessinsider.com/guides/tech/what-is-a-dns-server
22. Figure 22. Positive feedback loop (left). Negative feedback loop (right). Difference between Negative feedback and Positive feedback. https://www.rfwireless-world.com/Terminology/Negative-feedback-vs-Positive-feedback-in-amplifier.html
23. Figure 23. Operational amplifier (opamp). Operational Amplifier Basics, Types and Uses, Article, MPS. https://www.monolithicpower.com/en/operational-amplifiers
24. Figure 24. Op-amps configured with diodes or resistors in their feedback loop to do: logarithmation, summation or product. Analog Multiplier – Circuitlab. https://www.circuitlab.com/circuit/wn63688zgbys/analog-multiplier/
25. Figure 25. NOT gate ("inverter") made of one pMOS (top) and one nMOS (bottom) transistor. Why do CMOS NOT gate designs differ from BJT NOT gate designs? https://electronics.stackexchange.com/questions/570389/why-do-cmos-not-gate-designs-differ-from-bjt-not-gate-designs
26. Figure 26. ANSI / IEC [4] (right) and MIL-STD-806B [5] (left) symbols foremost common 7/16 dual-input logic gates (elementary), with their names. Logic Gates. https://learnabout-electronics.org/Digital/dig21.php
27. Figure 27. The truth tables for the most commonly used logic gates. Nucleic Acid Computing and its Potential to Transform Silicon0Based Technology. https://www.researchgate.net/publication/291418819_Nucleic_Acid_Computing_and_its_Potential_to_Transform_Silicon-Based_Technology
28. Figure 28. The 74HC00 circuit, a quad-input NAND circuit in DIP14 package. 74HC00 Quad 2-Input Nand Gate. https://www.technobotsonline.com/74hc00-quad-2-input-nand-gate.html
29. Figure 29. The 74HC04 circuit, a hex-inverter circuit in a 14-pin package (DIP14 or SOIC14). 74HC04 Hex Inverters. https://www.technobotsonline.com/74hc00-quad-2-input-nand-gate.html
30. Figure 30. Output requirements vs. input requirements. Logic Level Shifters for Driving LED Strips – Electric Fire Design. https://electricfiredesign.com/2021/03/12/logic-level-shifters-for-driving-led-strips/
31. Figure 31. Color-coded six-band THT resistor, and the table to decode it. Four bands are for resistance, one for tolerance and one for temperature coefficient. Resistor Calculator. https://www.calculator.net/resistor-calculator.html
32. Figure 32. Examples for SMD resistors of various resistance. Working Principle And Color Code Of Resistor. https://circuitspedia.com/resistor-working-types-of-resistor-color-code/
33. Figure 33. LM317 adjustable regulator configuration via R2 variable resistor. LM317 datasheet. https://www.ti.com/product/LM317
34. Figure 34. Voltage divider. Voltage Divider Calculator, https://www.allaboutcircuits.com/tools/voltage-divider-calculator/
35. Figure 35. A 3.3 Ohm THT resistor manufactured to be able to disspate up to 100W. Rezistor 3.3R, 100W, https://www.sigmanortec.ro/rezistor-33r-100w

36. Figure 36. Pictures of different types of resistors. Whare are Resistorrs? Types of Resistors and Their Uses. https://engineeringlearn.com/what-are-resistors-types-of-resistors-and-their-uses-complete-details-with-pictures/
37. Figure 37. Various types of capacitors. Factors at Play When Choosing the Right Capacitor for Your Design. https://www.ultralibrarian.com/2022/07/21/factors-at-play-when-choosing-the-right-capacitor-for-your-design-ulc
38. Figure 38. The RC integrator. Vo increases to Vi after infinite time. Theory of RC integrator. http://www.evalidate.in/lab1/pages/RC/RCIntegrator/RCIntegrator_T.html
39. Figure 39. RC low-pass filter Bode plots (for gain and phase). Passive Low Pass Filter, https://www.electronics-tutorials.ws/filter/filter_2.html
40. Figure 40. The RC time constant in RC integrator circuit. RC Charging Circuit Tutorial & RC Time Constant. https://www.electronics-tutorials.ws/rc/rc_1.html
41. Figure 41. Various types of inductorshttps://neosid.de/en/products/inductors/
42. Figure 42. Various diodes formats and types https://components101.com/articles/introduction-to-different-types-of-diodes
43. Figure 43. U/I plot for various diodes with their reverse voltages and maximum temperature. Schottky barrier diode (SBD)? https://toshiba.semicon-storage.com/ap-en/semiconductor/knowledge/faq/diode/what-are-schottky-barrier-diodessbds.html
44. Figure 44. The current as a function of forward voltage and temperature in RB451UM Schottky diode. RB451UM:Diode. https://fscdn.rohm.com/en/products/databook/datasheet/discrete/diode/schottky_barrier/rb451um-e.pdf
45. Figure 45. The RPM7140 infrared receiver: internal block schematic: amplifier, limiter, band-pass filter, detector and comparator. Output is open collector with 22kOhm integrated pull-up resistor. RPM7140, https://store.comet.bg/download-file.php?id=1668
46. Figure 46. The RPM7140's output signal (down) as reaction to input light (up). RPM7140, https://store.comet.bg/download-file.php?id=1668
47. Figure 48. Zener diode's operating regions. What is a Zener Diode? TTI Inc, https://www.tti.com/content/ttiinc/en/resources/blog/what-is-a-zener-diode.html
48. Figure 47. Various color temperatures and their equivalence in real-world. How to fix color cast https://lightroomguy.com/how-to-fix-color-cast-in-lightroom-classic/
49. Figure 49. Types of transistors, Transistor Tutorial about Bipolar and FET transistors, https://www.electronics-tutorials.ws/transistor/tran_8.html
50. Figure 50. Common usages for BJT: line driver, open-collector and push-pull, Choosing Between Line Driver, Open Collector, and Push Pull Encoders for NI Device – NI, https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019MXOSA2&l=en-US
51. Figure 51. Darlington transistors. The TR1 and TR2's amplification are multiplied. Darlington Transistor and the Szikilai Darlington Pair, https://www.electronics-tutorials.ws/transistor/darlington-transistor.html
52. Figure 52. Push-pull stage with FET transistors, usually found in output pads of microcontrollers, Open Drain Output vs. Push-Pull Output – Open4Tech, https://open4tech.com/open-drain-output-vs-push-pull-output/
53. Figure 53.Half-duplex communication between STM32 and another device. HOWTO: Use STM32 SPI half duplex mode – Digital Me, https://ba0sh1.com/2014/05/31/howto-use-stm32-spi-half-duplex-mode/
54. Figure 54. Types of switches. Different Types of Switches with Circuits and Applications, https://www.watelectronics.com/types-of-switches-with-applications/

55. Figure 55. Contact bouncing problem. L3: Debouncing - Physical Computing. https://makeabilitylab.github.io/physcomp/arduino/debouncing.html
56. Figure 56. Tactile switches, SMD (top left) or THT. Tactile Switches 101 – CUI Devices. https://www.cuidevices.com/blog/tactile-switches-101
57. Figure 57. Example of 8-slot DIP-switch. ECE EGD/EDS SERIES SLIDE TYPE, https://www.endrich.com/sixcms/media.php/2/%20EDS.pdf
58. Figure 58. Internal construction of a DIP switch. ECE EGD/EDS SERIES SLIDE TYPE, https://www.endrich.com/sixcms/media.php/2/%20EDS.pdf
59. Figure 59. Hearing for humans (source: NIH/NIDCD). How do we hear? NIDCD. https://www.nidcd.nih.gov/health/how-do-we-hear
60. Figure 60. Hearing loss in humans, depending on sex and age. Kim, Sunghee & Lim, Eun & Kim, Hak & Park, Jun & Jarng, Soon & Lee, Sang. (2010). Sex Differences in a Cross Sectional Study of Age-related Hearing Loss in Korean. Clinical and experimental otorhinolaryngology. 3. 27-31. 10.3342/ceo.2010.3.1.27.
61. Figure 61. Examples of noises a human may perceive. https://decibelpro.app/blog/what-is-50-decibels/
62. Figure 62. Overview of dB key values and their conversion into power and voltage ratios. Caspers, Fritz. (2012). RF measurements I: signal receiving techniques.a
63. Figure 63. The Fourier series of a function f(x). Fourier Series – from Wolfram MathWorld. https://mathworld.wolfram.com/FourierSeries.html
64. Figure 64. Decomposition of a square wave into first four terms of Fourier series. Fourier Series – from Wolfram MathWorld. https://mathworld.wolfram.com/FourierSeries.html
65. Figure 65. Formula for observed frequency (in Doppler effect). Doppler Effect. https://www.guyhowto.com/doppler-effect/
66. Figure 66. Doppler effect: as the source and observer move closer, the distance between successive wave crests decreases, resulting in a rise in perceived frequency; the opposite is the case as the two move apart. (Science Facts). Using Doppler Effect to Determine Object Motion. https://www.digikey.ro/ro/blog/the-doppler-effect-now-widely-accepted-and-easy-to-use
67. Figure 67. The electromagnetic wave. http://hyperphysics.phy-astr.gsu.edu/hbase/Waves/emwavecon.html
68. Figure 68. The electromagnetic spectrum. Mini Physics. https://www.miniphysics.com/electromagnetic-spectrum_25.html
69. Figure 69. Filter types and their responses. Understanding filtering basics. https://www.vectornav.com/resources/inertial-navigation-primer/math-fundamentals/math-filtering
70. Figure 70. First-order low-pass filter frequency response with gain and phase (Bode plots). Electronic applications: 2.6 The full Bode plor:gain and phase. https://www.open.edu/openlearn/science-maths-technology/electronic-applications/content-section-2.6
71. Figure 71. The RMS value of X. Easy Formula RMS Voltage and Current AC Circuit | Wira Electrical, https://wiraelectrical.com/rms-voltage-and-current-ac/
72. Figure 72. The Vrms, Vpeak and Vpeak-to-peak values of a sine AC voltage. RMS Voltage Calculator, https://circuitdigest.com/calculators/rms-voltage-calculator
73. Figure 73. A full wave AC rectifier. https://passive-components.eu/ripple-current-and-its-effects-on-the-performance-of-capacitors/
74. Figure 74. 230VAC-to-5V, transformer-based power supply based on a 7805 IC. 5Volt Regulator Circuit Diagram. https://www.circuits-diy.com/7805-5-volt-regulator-circuit-diagram/

75. Figure 75. Schematic of a modern PSU with possible component selection. https://eu.mouser.com/new/bourns/bourns-power-supply
76. Figure 76. Step-down AC transformer. https://www.etechnog.com/2021/07/transformer-diagram-and-constructional.html
77. Figure 77. Series-voltage regulator based on a transistor and Zener diode. Transistor Series Voltage Regulator. https://www.watelectrical.com/what-is-transistor-series-voltage-regulator-working-and-its-experiment/
78. Figure 78. The voltage doubler. The fundamentals of a Charge Pump Circuit. https://www.allaboutcircuits.com/technical-articles/switched-capacitor-circuits-charge-pump-circuit-basics/
79. Figure 79. The buck converter topology. DC to DC Buck converter tutorial & diagram | Analog Devices. https://www.analog.com/en/technical-articles/dc-to-dc-buck-converter-tutorial.html
80. Figure 80. The boost converter topology. DC to DC Buck converter tutorial & diagram | Analog Devices. https://www.analog.com/en/technical-articles/dc-to-dc-buck-converter-tutorial.html
81. Figure 81. Buckboost in charging stage. Buck-Boost Converter. https://www.electrical4u.com/buck-boost-converter/
82. Figure 82. Buckboost in discharging stage. Buck-Boost Converter. https://www.electrical4u.com/buck-boost-converter/
83. Figure 83. Typical MC34063 usage as step-up converter. Additional output filters may be added to smooth the output. https://www.ti.com/lit/ds/symlink/mc34063a.pdf
84. Figure 84. GP's selection guide for cylindrical primary battery types. https://ind.gpbatteries.com/pub/media/uploads/pdf/product_catalogues/180309_IND_CTL-PB_INT_Catalog-Selection-Guide-Preview.pdf
85. Figure 85. CR2032 coin cell BIOS-backing battery, having 20 mm diameter and 3.2 mm thickness. https://www.murata.com/products/productdata/8811861704734/CR2032-SPECSHEET.pdf
86. Figure 86. Cycle life vs depth of discharge (left). Storage time vs temperature. http://media.vitacom.ro/pictures/pdf/b/Acumulator%20plumb%20acid%20cu%20gel%20Ultracell%2012V%20100AH.pdf
87. Figure 87. Effect of discharge current, on capacity in UCG100-12 lead-acid battery. http://media.vitacom.ro/pictures/pdf/b/Acumulator%20plumb%20acid%20cu%20gel%20Ultracell%2012V%20100AH.pdf
88. Figure 88. 4S3P pack. Note the white top part and the grey top part. https://voltaplex.com/4s3p-14.4v-7.5ah-li-ion-18650-battery-pack-samsung-25r5-cuboid
89. Figure 89. Toyota Prius 2 Hybrid battery: 28 modules of 6 cells each having 1.2V and 6.5Ah NiMH. Total voltage: 28*6*1.2 = 201.6 V nominal. Total pack energy: 201.6 V * 6.5 Ah = 1.3 kWh. https://www.ebay.com/itm/185569968246
90. Figure 90. Snapshot of Dr.Prius app, which reads voltages across all 14 battery packs of Prius 3, in order to diagnose “bad” cells. https://play.google.com/store/apps/details?id=com.nexcell.app&hl=en_US
91. Figure 91. Very common laptop-battery element (li-Ion 18650 format). https://www.tme.eu/en/details/accu-inr18650-35e/rechargeable-batteries/samsung-sdi/inr18650-35e-as/
92. Figure 92. A typical algorithm for charging Li-Ion batteries. Constant current, then constant voltage. https://www.digikey.com/en/maker/blogs/charging-lithium-ion-batteries
93. Figure 93. NiCd and NiMH dv/dt and dT/dt marking the end of charge. https://www.mpoweruk.com/chargers.htm
94. Figure 94. Common physical formats for batteries. https://power.bigbadmole.com/en/batarejki/vidy.html

95. Figure 95. Summary for specific energy and specific power across all common battery chemistries. Lyubimov, I & Kurmaev, R. (2020). The choice of a performance criterion for a high-voltage battery of a vehicle. IOP Conference Series: Materials Science and Engineering. 819. 012015. 10.1088/1757-899X/819/1/012015.
96. Figure 96. Data transmission types. Simplex vs. Duplex Fibre. https://www.black-box.de/en-de/page/25078/Resources/Technical-Resources/Black-Box-Explains/Fibre-Optic-Cable/simplex-vs-duplex-fiber-patch-cable
97. Figure 97. Parallel vs Serial interfaces. Fundamentals of communication and networking. https://bournetocode.com/projects/AQA_A_Theory/pages/3-9.html
98. Figure 98. Example of a SPI communication (8-bit per frame). https://alchitry.com/serial-peripheral-interface-spi-verilog
99. Figure 99. START and STOP conditions on I2C bus. https://www.nxp.com/docs/en/user-guide/UM10204.pdf
100. Figure 100 Master sends data to slave (write frame) with 7-bit address. https://www.nxp.com/docs/en/user-guide/UM10204.pdf
101. Figure 101. Master reads data from slave, with 7-bit address (read frame). https://www.nxp.com/docs/en/user-guide/UM10204.pdf
102. Figure 102. Two partners exchanging data over a full-duplex UART bus. GND is tied together, but not shown. https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html
103. Figure 103. UART data frame https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html
104. Figure 104. A UART data frame with START bit, 8 data bits, one parity bit and STOP bit. https://developer.electricimp.com/resources/uart
105. Figure 105. Von Neumann and Harvard Architectures. https://www.watelectronics.com/difference-between-von-neumann-and-harvard-architecture/
106. Figure 106. PIC10F2xx internal block diagram. https://ww1.microchip.com/downloads/en/DeviceDoc/40001239F.pdf
107. Figure 107. Clamping diodes protect the I/O on the left https://www.digikey.com/en/articles/protecting-inputs-in-digital-electronics
108. Figure 108. Prescaler input (top), and output for 1:2, 1:4 or 1:8 setting https://wiki.dcae.pub.ro/index.php/CID_aplicatii_9_:_Numaratorul
109. Figure 109. MSP430 MCU timer modes of operation http://www.ocfreaks.com/msp430-timer-programming-tutorial/
110. Figure 110. Brown-out detector scenario https://microchipdeveloper.com/8avr:bod
111. Figure 111 N-bit SAR ADC architecture https://www.analog.com/en/technical-articles/successive-approximation-registers-sar-and-flash-adcs.html
112. Figure 112 R2R DAC architecture https://www.tek.com/en/blog/tutorial-digital-analog-conversion-r-2r-dac
113. Figure 113. AMD/Xilinx Zynq 7000 SoC chip family. https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html
114. Figure 114. Intel/Altera Cyclone V SoChttps://www.rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoard
115. Figure 115. NVIDIA Jetson Nano SoC. https://developer.nvidia.com/blog/solving-entry-level-edge-ai-challenges-with-nvidia-jetson-orin-nano/
116. Figure 116. SOIC-14-N (narrow) package, having P (pin-to-pin distance) of 1.27mm (that is 0.05 inch, or 50 mils). SOIC14. https://toshiba.semicon-storage.com/ap-en/semiconductor/design-development/package/detail.SOIC14.html

117. Figure 117. DIP14 package, having pin-to-pin distance of 2.54 mm (0.1 inch). DIP14. https://toshiba.semicon-storage.com/ap-en/semiconductor/design-development/package/detail.DIP14.html
118. Figure 118. A PCB (printed circuit board) with multiple SMD components. Closeup of computer parts. https://www.freepik.com/free-photo/closeup-computer-parts_5926578.htm
119. Figure 119. A PCB with SMD capacitors. Transistors. https://www.freepik.com/free-photo/transistors_906596.htm
120. Figure 120. A PCB with multiple components in THT (diodes, resistors, ICs). Top view circuit board close-up. https://www.freepik.com/free-photo/top-view-circuit-board-close-up_20282387.htm
121. Figure 121. A 3D rendering of a simplified PC motherboard. System plate for pc isometric illustration with semiconductor elements slots. https://www.freepik.com/free-vector/system-plate-pc-isometric-illustration-with-semiconductor-elements-slots-microchips-capacitors-diodes-transistors_7496526.htm
122. Figure 122. A comparison of speed of intensive computation across multiple languages. C/C++ is fastest when compiled with optimizing compiler (gcc). https://github.com/niklas-heer/speed-comparison
123. Figure 123. Representing a g(n) function which asymptotically dominate f(n) function. Big-O notation. https://xlinux.nist.gov/dads/HTML/bigOnotation.html
124. Figure 124. The O-notation complexity increase with the number of elements processed. O(1) and O(logn) are usually excellent complexities, O(n) is fair, and O(n*logn) is usually considered almost decent in algorithms and circuits Big-O Algorithm Complexity Cheat Sheet (Know Thy Complexities!) @ericdrowell. https://www.bigocheatsheet.com/
125. Figure 125. Matrix-matrix multiplication (naïve-approach). CUDA C++ Programming Code. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
126. Figure 126. Caesar cipher exemplified, for a distance of 3. Caesar Cipher in Cryptography. https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/
127. Figure 127. DES encryption where F is a Feistel one-way . CS 463 Lecture. https://www.cs.uaf.edu/2013/spring/cs463/lecture/02_06_Feistel.html
128. Figure 128. One round of DES encryption, Sx are the 6-bit to 4-bit substitution boxes. What is 3DES encryption and how does DES work? | Comparitech. https://www.comparitech.com/blog/information-security/3des-encryption/
129. Figure 129. One AES round, expressed graphically in 3D. AES_Roound_Function, https://upload.wikimedia.org/wikipedia/commons/5/50/AES_%28Rijndael%29_Round_Function.png
130. Figure 130. Left is the original image, middle is ECB mode cryptotext and right is CTR. Linux2.0 Penguins, https://isc.tamu.edu/~lewing/linux/, lewing@isc.tamu.edu Larry Ewing and The GIMP, https://commons.wikimedia.org/wiki/File:Tux_ECB.png, https://upload.wikimedia.org/wikipedia/commons/5/58/Tux_secure.png
131. Figure 131. Top is ECB encryption and bottom is ECB description: great for parallel processing but very weak in hiding data. ECB encryption. https://upload.wikimedia.org/wikipedia/commons/d/d6/ECB_encryption.svg https://upload.wikimedia.org/wikipedia/commons/e/e6/ECB_decryption.svg
132. Figure 132. Top is CBC encryption and bottom is CBC decryption. CBC is hard to parallelize in encryption, but it is suitable for parallel decryption. https://commons.wikimedia.org/wiki/File:CBC_encryption.svg, https://commons.wikimedia.org/wiki/File:CBC_decryption.svg
133. Figure 133Figure 133. CTR mode encryption (top) and decryption (bottom).

References

1. https://www.iso.org/standard/31898.html accessed on 10.08.2023
2. https://www.jedec.org accessed on 10.08.2023
3. Loops & Complexity in DIGITAL SYSTEMS (Lecture notes on Digital Design in Ten Giga-Gate/Chip Era) by Gheorghe M. Stefan, link: http://users.dcae.pub.ro/~gstefan/2ndLevelteachingMaterials/0-BOOK.pdf accessed on 25.07.2023
4. "IEEE Standard Graphic Symbols for Logic Functions (Including and incorporating IEEE Std 91a-1991, Supplement to IEEE Standard Graphic Symbols for Logic Functions)," in *IEEE Std 91a-1991 & IEEE Std 91-1984,* vol., no., pp.1-160, 13 July 1984, doi: 10.1109/IEEESTD.1984.7896954.
5. MIL-STD-806B, https://bitsavers.org/pdf/mil-std/MIL-STD-806B_Graphical_Symbols_For_Logic_Diagrams_19620226.pdf accessed on 28.07.2023
6. Abels, Seth & Khisamutdinov, Emil. (2015). Nucleic Acid Computing and its Potential to Transform Silicon-Based Technology. DNA and RNA Nanotechnology. 2. 10.1515/rnan-2015-0003.
7. C99, ISO/IEC 9899:1999 standard, https://www.iso.org/standard/29237.html accessed on 30.07.2023
8. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd. ed.). The MIT Press.
9. Williams, V.V., Xu, Y., Xu, Z., & Zhou, R. (2023). New Bounds for Matrix Multiplication: from Alpha to Omega. ArXiv, abs/2307.07970.
10. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html accessed on 7.08.2023
11. https://www.top500.org/ accessed on 8.08.2023
12. https://csrc.nist.gov/files/pubs/fips/81/final/docs/fips81.pdf accessed on 9.08.2023
13. https://www.tiobe.com/tiobe-index/ accessed on 10.08.2023
14. I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, Croatia, 2012, pp. 1725-1730.
15. O. C. Novac, D. E. Madar, C. M. Novac, G. Bujdosó, M. Oproescu and T. Gal, "Comparative study of some applications made in the Angular and Vue.js frameworks," *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*, Oradea, Romania, 2021, pp. 1-4, doi: 10.1109/EMES52337.2021.9484150.
16. Greg Lim; Daniel Correa, *Django 4 for the Impatient: Learn the core concepts of Python web development with Django in one weekend* , Packt Publishing, 2022.
17. K. Saundariya, M. Abirami, K. R. Senthil, D. Prabakaran, B. Srimathi and G. Nagarajan, "Webapp Service for Booking Handyman Using Mongodb, Express JS, React JS, Node JS," *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, Coimbatore, India, 2021, pp. 180-183, doi: 10.1109/ICSPC51351.2021.9451783.
18. https://github.com/niklas-heer/speed-comparison accessed on 10.08.2023
19. DES encryption standard, https://csrc.nist.gov/files/pubs/fips/46/final/docs/nbs.fips.46.pdf accessed on 10.08.2023
20. AES encryption standard, FIPS197 url: https://doi.org/10.6028/NIST.FIPS.197-upd1 Accessed on 10.08.2023
21. https://home.cern/science/computing/birth-web/short-history-web
22. https://www.w3.org/
23. https://www.nidcd.nih.gov/news/multimedia/journey-of-sound-video, accessed on 15.08.2023
24. DecibelPro.App, https://decibelpro.app/blog/what-is-50-decibels/ accessed on 15.08.2023

25. Kim, Sunghee & Lim, Eun & Kim, Hak & Park, Jun & Jarng, Soon & Lee, Sang. (2010). Sex Differences in a Cross Sectional Study of Age-related Hearing Loss in Korean. Clinical and experimental otorhinolaryngology. 3. 27-31. 10.3342/ceo.2010.3.1.27.
26. Caspers, Fritz. (2012). RF measurements I: signal receiving techniques, https://arxiv.org/abs/1201.3247
27. Lyubimov, I & Kurmaev, R. (2020). The choice of a performance criterion for a high-voltage battery of a vehicle. IOP Conference Series: Materials Science and Engineering. 819. 012015. 10.1088/1757-899X/819/1/012015.
28. Software Arduino https://www.arduino.cc/en/software accessed 03.09.2023
29. C. Bira, A-M Calfa, M. Tudosie, Automatic Memory Wear Test for Flash-Based Microcontrollers, REV2011 Proceedings, 29 June – 2 July, pp291-295
30. Bîră, C.; Ionescu, L.; Rusu-Casandra, A. The Radar Signal Processor of the First Romanian Space Surveillance Radar. Remote Sens. 2023, 15, 3630. https://doi.org/10.3390/rs15143630
31. A. Manikas, EE3-27: Principles of Classical and Modern Radar, https://skynet.ic.ac.uk/notes/Radar_6_CW_Radar.pdf
32. L.Ionescu, V.Turcu, D.Istriteanu, R.Scagnoli, Cheia antenna retrofit, NEOSST1 (1st NEO and Debris Detection Conference), 2019, ESA Space Safety Programme Office https://conference.sdo.esoc.esa.int/proceedings/neosst1/paper/415
33. Ionescu, L.; Rusu-Casandra, A.; Bira, C.; Tatomirescu, A.; Tramandan, I.; Scagnoli, R.; Istriteanu, D.; Popa, A.-E. Development of the Romanian Radar Sensor for Space Surveillance and Tracking Activities. Sensors 2022, 22, 3546. https://doi.org/10.3390/s22093546
34. ATS9462-16 bit, https://www.alazartech.com/en/product/ats9462/13/ accessed on 20.09.2023
35. Pisanu, T.; Muntoni, G.; Schirru, L.; Ortu, P.; Urru, E.; Montisci, G. Recent Advances of the BIRALET System about Space Debris Detection. Aerospace 2021, 8, 86. https://doi.org/10.3390/aerospace8030086
36. Pandeirada, J.; Bergano, M.; Neves, J.; Marques, P.; Barbosa, D.; Coelho, B.; Ribeiro, V. Development of the First Portuguese Radar Tracking Sensor for Space Debris. Signals 2021, 2, 122–137
37. Casado Gómez, R.; Martínez-Villa Salmerón, J.; Besso, P.; Alessandrini, M.; Pinna, G.; Prada, M. Initial operations of the breakthrough Spanish Space Surveillance and Tracking Radar (S3TSR) in the European context. In Proceedings of the 1st NEO and Debris Detection Conference, Darmstadt, Germany, 22–24 January 2019; Available online: https://conference.sdo.esoc.esa.int/proceedings/neosst1/paper/479 (accessed on 20 February 2023).
38. Space Observation Radar TIRA [Online]. Available online: https://www.fhr.fraunhofer.de/en/the-institute/technical-equipment/Space-observation-radar-TIRA.html (accessed on 27 May 2023).
39. GESTRA Space Radar Passes Its First Test, [Online]. Available online: https://www.dlr.de/en/latest/news/2019/04/20191129_latest-radar-technology (accessed on 27 May 2023).
40. GESTRA [Online]. Available online:https://www.radartutorial.eu/19.kartei/02.surv/karte075.en.html (accessed on 27 May 2023).
41. Michal, T.; Eglizeaud, J.P.; Bouchard, J. GRAVES: The new French System for Space Surveillance. In Proceedings of the Fourth European Conference on Space Debris, Darmstadt, Germany, 18–20 April 2005

42. Jon D. Eastment, ASSESSING THE SUITABILITY OF THE CHILBOLTON RADARS FOR SPACE DEBRIS AND SPACE SURVEILLANCE APPLICATIONS, 2009 ESA Conference 5th European Conference on Space Debris, Volume 5, Issue 1
43. Goddard, J. W. F. et al. “The Chilbolton advanced meteorological radar: a tool for multidisciplinary atmospheric research.” Electronics & Communication Engineering Journal 6 (1994): 77-86.
44. Vizitiu C, Bîră C, Dinculescu A, Nistorescu A, Marin M. Exhaustive Description of the System Architecture and Prototype Implementation of an IoT-Based eHealth Biometric Monitoring System for Elders in Independent Living. Sensors (Basel). 2021 Mar 6;21(5):1837. doi: 10.3390/s21051837. PMID: 33800728; PMCID: PMC7961703.
45. A. Dinculescu, C. Dugan, C. Vizitiu and I. Parlatescu, "Study on Choice Reaction Time as a Complementary Method in Idiopathic Orofacial Pain," 2021 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 2021, pp. 1-4, doi: 10.1109/EHB52898.2021.9657580.

Călin Bîră, PhD., is Associate Professor at the Politehnica University of Bucharest, where he teaches undergraduate and graduate courses related to programming, microcontrollers, digital design, signal acquisition and processing. He obtained his PhD in 2013 from the Politehnica University of Bucharest, with a thesis on programming environment for (energy-efficient embedded) accelerators, thus gradually shifting to an academic career. His work with the Faculty of Electronics, Telecommunications and Information Technology aims to bridge academia (research and teaching) with business and industry and their ways of designing research and development projects, thus addressing the gap between university and businesses. Călin Bîră's commitment to academic research and strengthening international collaboration and teamwork is made visible by multiple research grants and projects (DocInvest, SAVE, DEXTER, ATLAS) as well as by publications in the domain of electronics, programming and signal processing. Prior to his pursuing an academic career, Călin Bîră worked for 8 years as an engineer for global companies, specializing in low-level software and hardware design. His current research interests include energy-efficient computation and embedded systems for research and commercial projects.