UiT Norges arktiske universitet

DTE- 2804: E-health and Systems

Lecture 1. Introduction to Embedded Systems and E-health systems

Arthur Schuchter
UiT – Campus Bodø

Autumn Term 2023

Introduction

1. Introduction to embedded systems
2. Memory types

3. single-board computers
3.1 Beaglebone Black (BBB)
3.2. Arduino
3.3. Rasberry Pi

4. Application of embedded systems

5. Coding and compression of data
5.1 Why do we still need coding and compression
5.2 Lossless coding and compression
5.2.1 arithmetic coding
5.2.2 Huffman coding
5.2.3 Runlength coding
5.2.4 Dictionary coding
5.3 Lossy coding and compression
5.3.1 Quantization
5.3.2 Transform coding

6. Understanding Digital Imaging

7. Medical imaging
7.1 MRI
7.2 CT
7.3 Differences
7.4 Ultrasounds

8. Medical Products and Companies
8.1 Medpresence

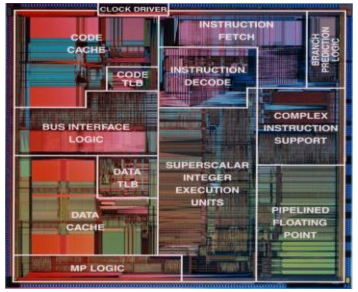
1: Introduction Embedded Systems

Embedded versus Single Board Computer

- A general-purpose computer such as a desktop or laptop can be used for a variety of applications (e.g., word processing, streaming audio, DB)
- But what is an embedded system? What is a Single Board Computer? → A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. Single-board computers were made as demonstration or development systems, for educational systems, or for use as embedded computer controllers. Many types of home computers or portable computers integrate all their functions onto a single printed circuit board.

Why are embedded systems needed?
Parrallel computing

Instruction-level Parallelism

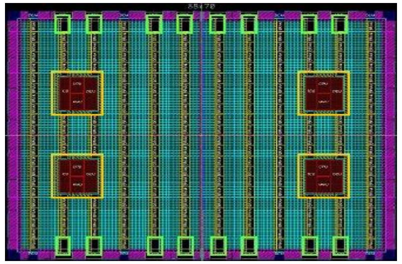


Superscalar

VLIW

Multi-threaded

Multi-Processing



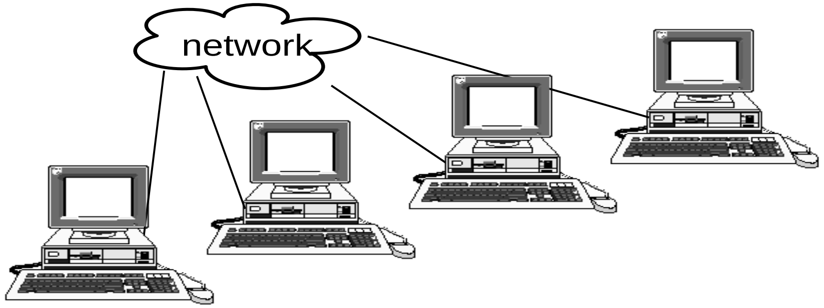
SMP

Multi-core FPGA

Microgrid

Parallel embedded system (SoC)

High Performance Computing

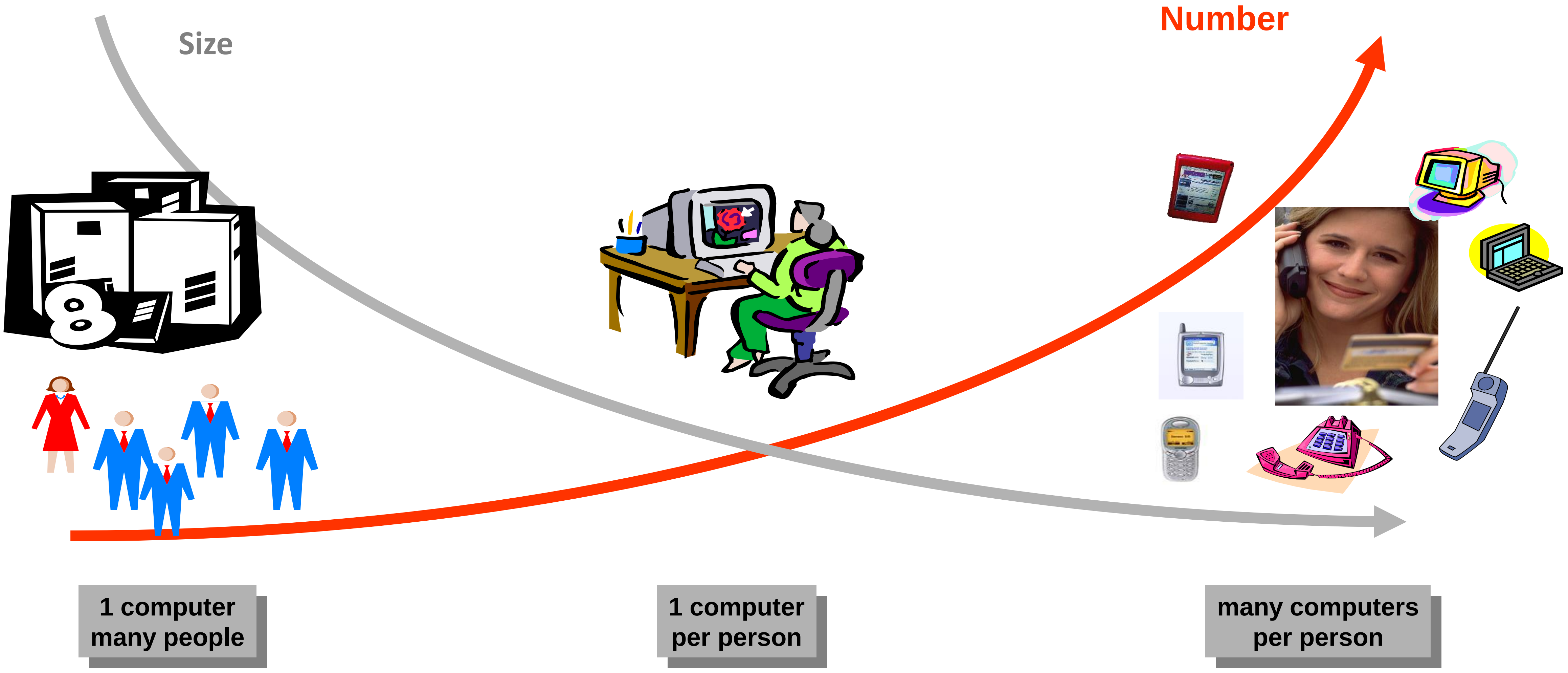


Cluster of workstations

Grid computing

Public resource computing

Embedded Systems Trend



Applications for embedded systems with many computers per person

Smart house

- Intelligent control of house functions
- Knows what you need when you get home

Smart car

- Adapts to different drivers, road conditions
- Gives advice on currently best routes
- Communicates with other cars on special events
- Integrates your personal devices into its network

Smart factory

- Intelligent control of production (federations of robots,…)
- Integrates supply chain management

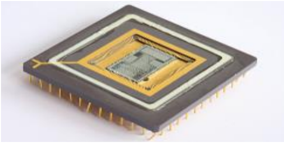
Embedded Computer System

- An embedded computer system is a component of a larger system
  - It generally interacts with the environment and performs a set of repeated control and monitoring tasks
  - There are a lot more embedded computing devices than desktops, laptops, workstations
- Embedded Processors - „embedd“ in larger devices“ such as
  - Microwave
  - Smartphone
  - Aeroplane, Car, House
  - Human?

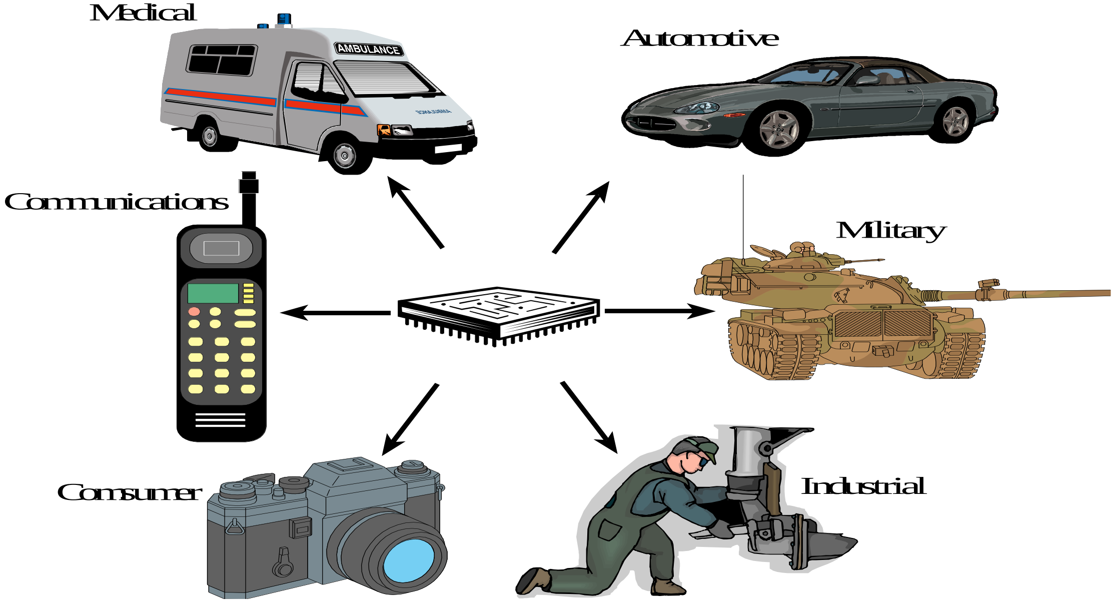

Efficiency in Embedded Systems

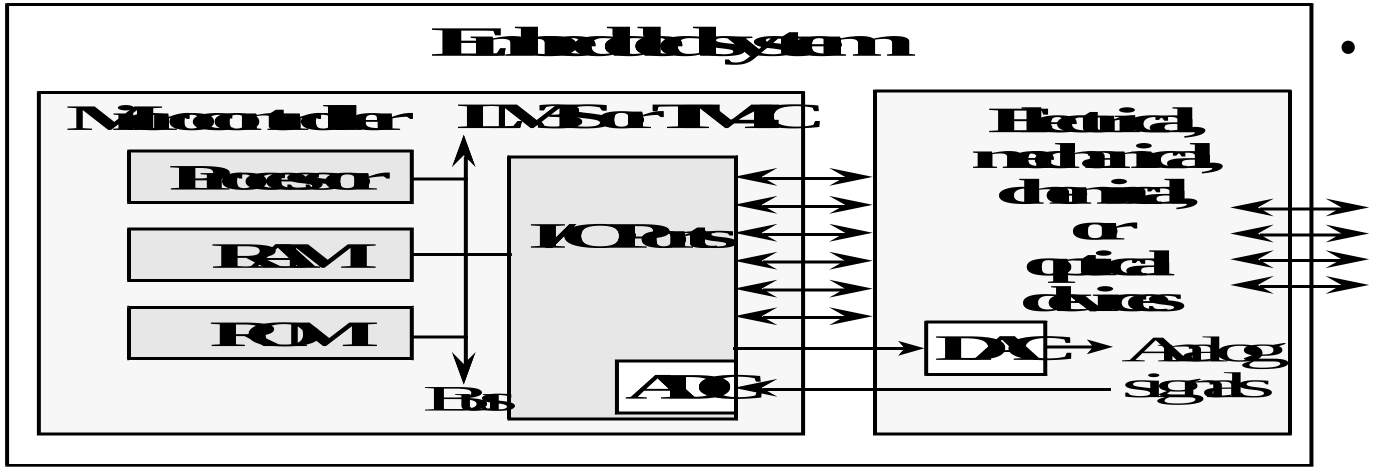
Codesize efficient

- Runtime efficient
- Weight efficient
- Cost efficient
- Energy efficient
- Real time
  - (not possible for every application with the Beaglebone)



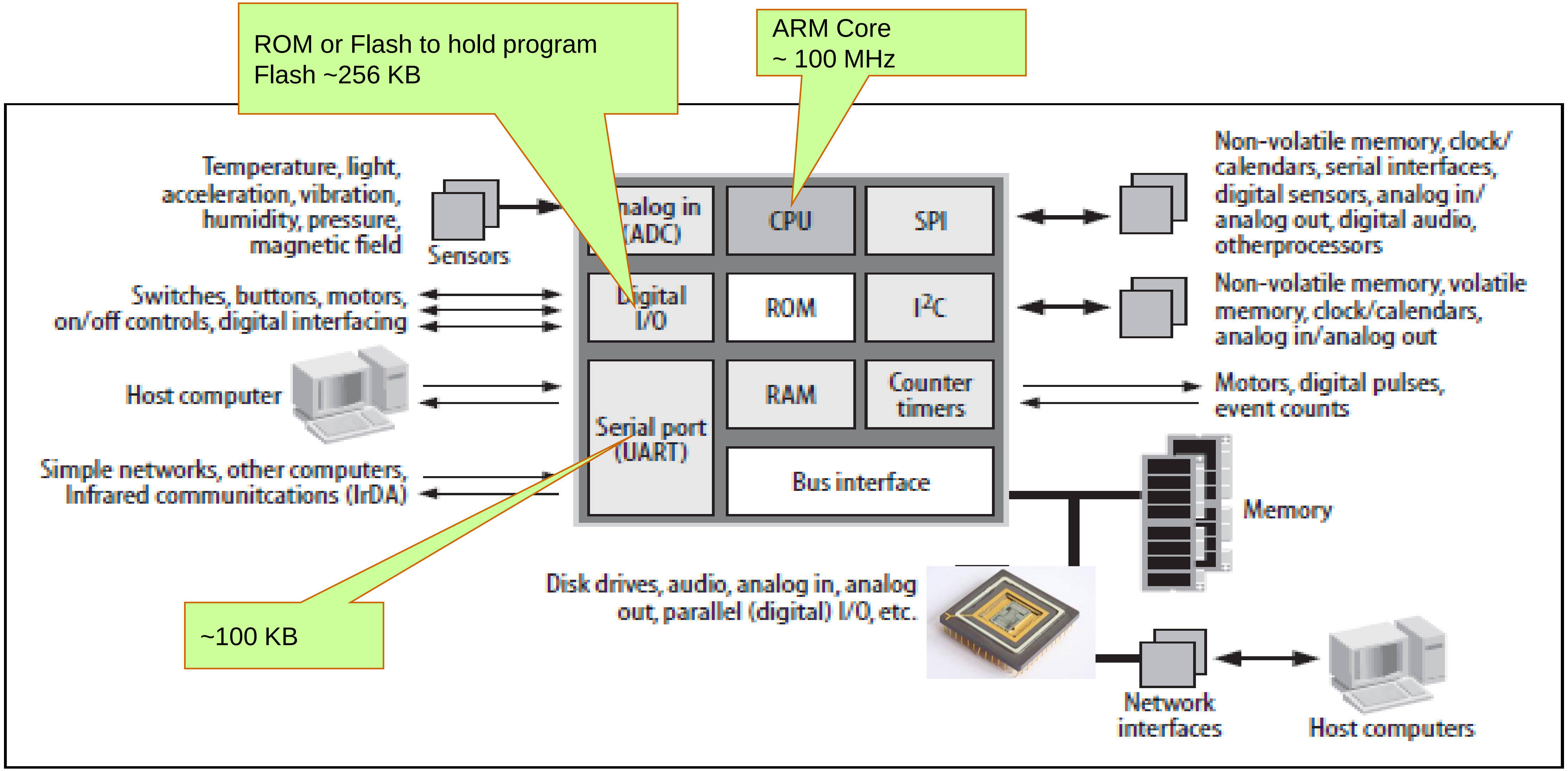
Embedded System





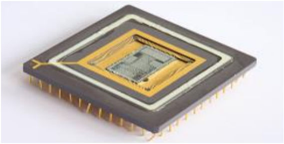
- Embedded Systems are everywhere
  - Ubiquitous, invisible
  - Hidden (computer inside)
  - Dedicated purpose
- Microprocessor
  - Intel: 4004, ..8080,.. x86
  - Freescale: 6800, .. 9S12,.. PowerPC
  - ARM, DEC, SPARC, MIPS, PowerPC, Natl. Semi.,…
- Microcontroller
  - Processor+Memory+
    I/O Ports (Interfaces)

Microcontroller-Based Embedded Computer



Efficency

- Real-time embedded systems must be efficient
- Code-size efficient
- Run-time efficient
- Weight efficient
- Cost efficient
- Energy efficient

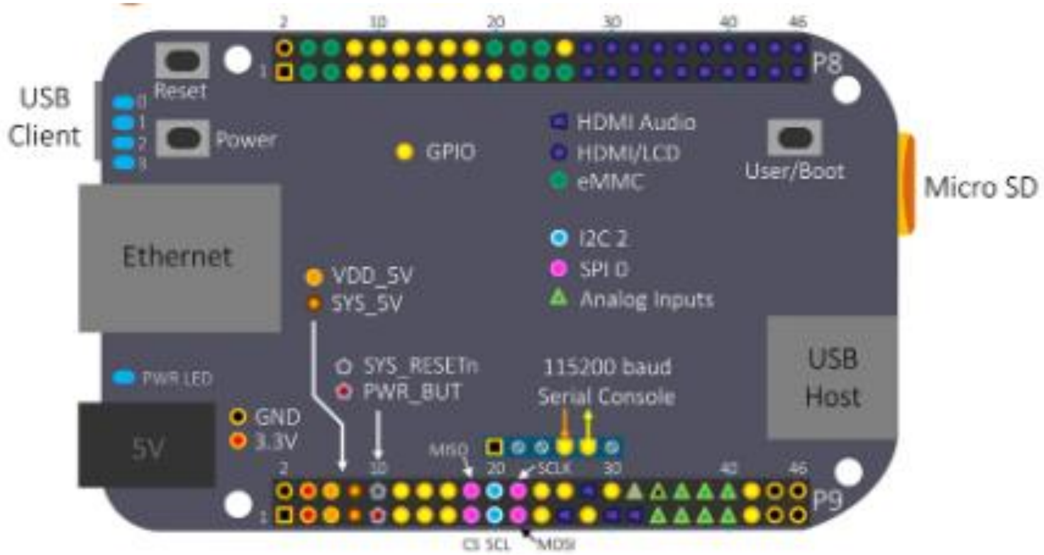


Microcontroller

- Processor – Instruction Set + memory + accelerators
  - Ecosystem
- Memory
  - Non-Volatile
    - ROM
    - EPROM, EEPROM, Flash
  - Volatile
    - RAM (DRAM, SRAM)
- Interfaces
  - H/W: Ports
  - S/W: Device Driver
  - Parallel, Serial, Analog, Time
- I/O
  - Memory-mapped vs. I/O-instructions (I/O-mapped)

General Purpose Input/Output (GPIO)

- is a generic pin on a chip whose behavior (including whether it is an input or output pin) can be controlled (programmed) by the user at run time.
- A general-purpose input/output (GPIO) is an uncommitted digital signal pin on an integrated circuit or electronic circuit board whose behavior—including whether it acts as an input or output—is controllable by the user at run time.



Universal asynchronous receiver-transmitter: UART

- The UART has a transmission engine, and also a reception engine (they can operate simultaneously)
- Software controls the UART's operations by accessing several registers, using the CPU's input and output instructions
- A universal asynchronous receiver transmitter (UART) is a computer hardware device for asynchronous serial communication. A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. The electric signaling levels and methods are handled by a driver circuit external to the UART.
- One or more UART peripherals are commonly integrated in microcontroller chips.
- A related device, the universal synchronous and asynchronous receiver transmitter (USART) also support synchronous operation.

2. Memory types

Memory Technologies

- DRAM: Dynamic Random Access Memory
  - updside: very dense (1 transistor per bit) and inexpensive
  - downside: requires refresh and often not the fastest access time
  - Often used for main memories
- SRAM: Static Random Access Memory
  - Upside: fast and no refresh requiredd
  - downside: not so dense and not so cheap
  - Often used for caches
- ROM: Read Only Memory
  - Often used for bootstrapping such

What is RAM?

- RAM is short for Random Access Memory, which is the main memory of the computer. This component is very important and it allows your computer to store data short-term for quicker access.
- When purchasing a computer, experienced users always ask "how much RAM does this computer have?" If you plan to use this computer to run a certain OS, play certain games, or run other apps, these programs always require your computer to meet a certain RAM requirement. Otherwise, these apps can't be run on the computer.

Differences among DRAM, SDRAM and SRAM

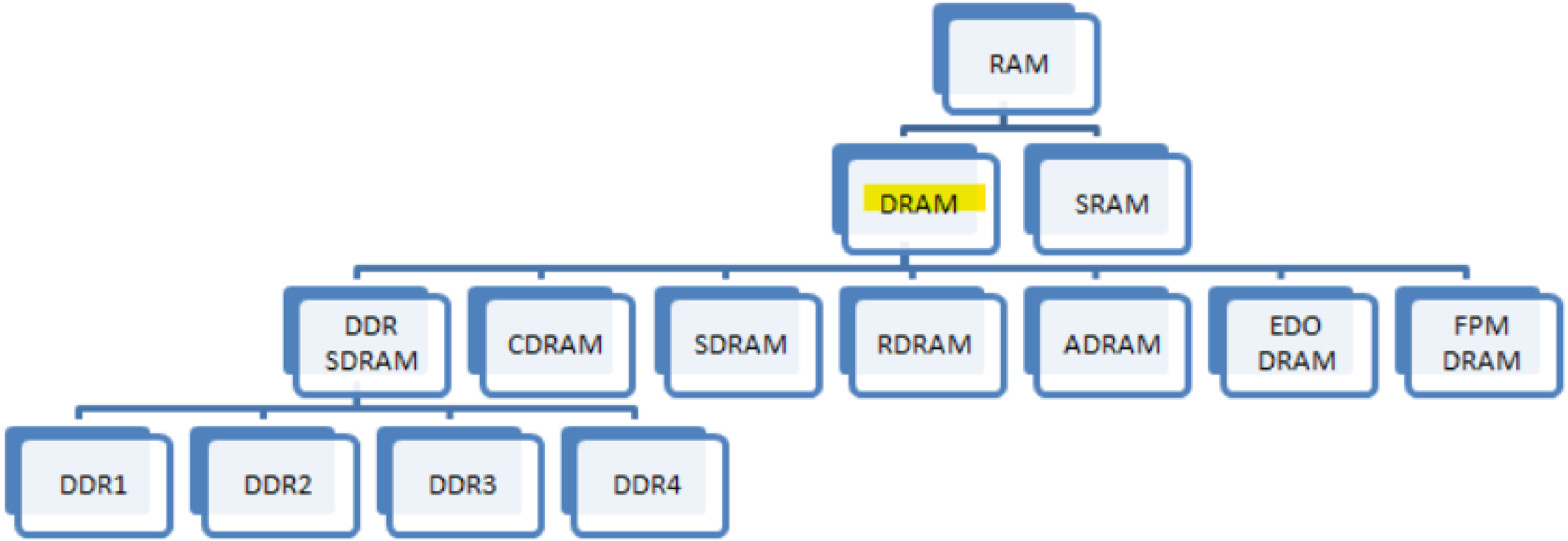
- SRAM: SRAM is short for static RAM without refreshing, whose speed is very fast. The cache inside the CPU belongs to static RAM. However, the disadvantage is that a memory unit requires a large number of transistors, so it is expensive in spite of its small capacity
- DRAM: Dynamic RAM requires refreshing, with a large capacity.
- SDRAM: Synchronous dynamic RAM need to refresh, with fast speed and large capacity
- DDR SDRAM: Dual-channel synchronous dynamic RAM, requires refreshing, with fast speed and large capacity.

Types of RAM: There are two main types

- SRAM: It requires about six transistors for a memory cell, which makes it high-cost, small-capacity, and high-speed. Therefore, it is often used as the primary cache or secondary cache of the CPU.
- DRAM: It requires about one transistor and one capacitor for a memory cell, which makes it low-cost and large-capacity, but it needs to be refreshed and the speed is slower than SRAM. Therefore, it is often used as the main memory of the computer.

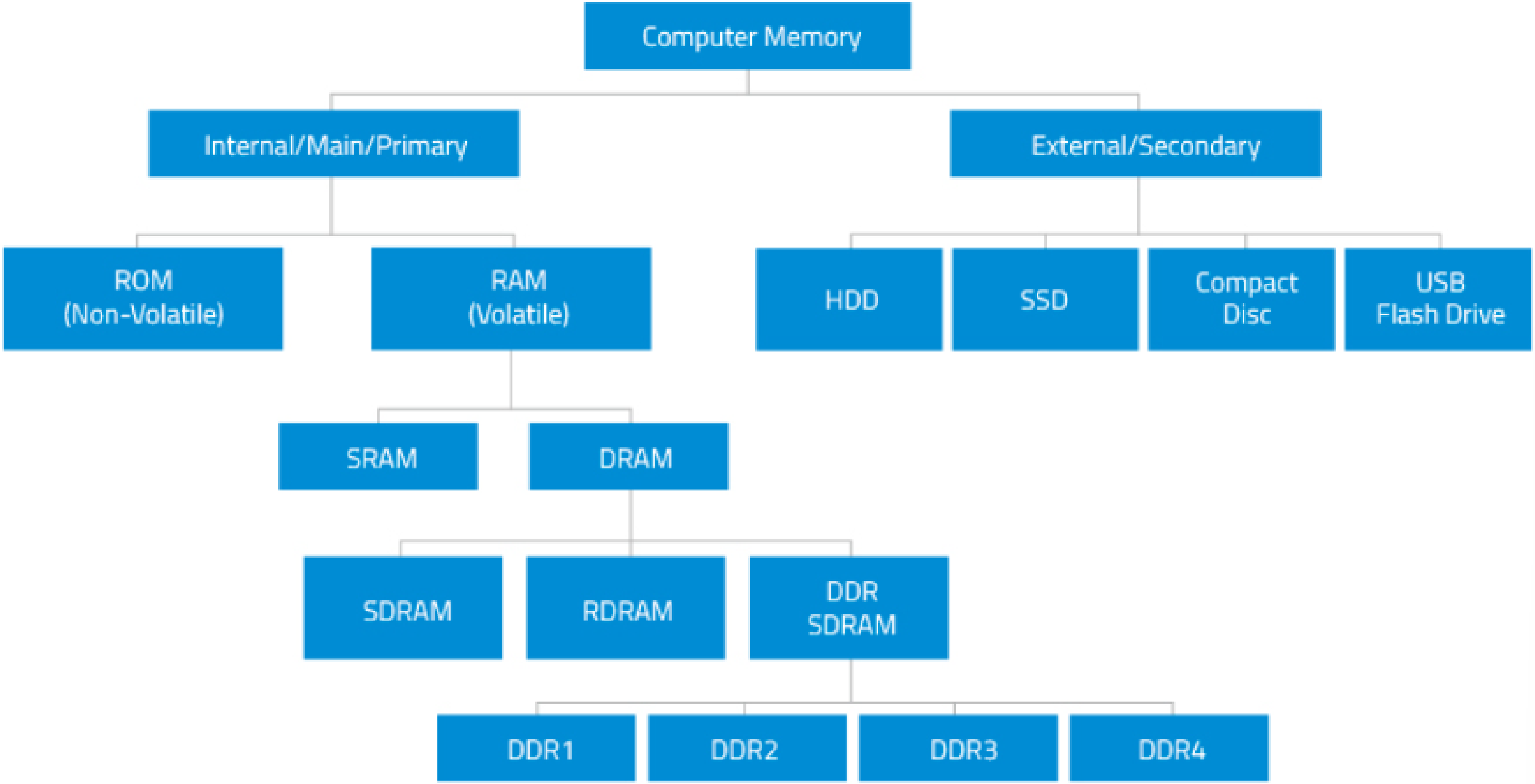
What is DRAM

- DRAM is called Dynamic RM. DRAM is widely used as a computer’s main memory. Each DRAM memory cell is made up of a transistor and a capacitor within an integrated circuit, and a data bit is stored in the capacitor. Since transistors always leak a small amount, the capacitors will slowly discharge, causing information stored in it to drain; hence, DRAM has to be refreshed (given a new electronic charge) every few milliseconds to retain data.



Common DRAM types

- Synchronous DRAM (SDRAM) “synchronizes” the memory speed with CPU clock speed so that the memory controller knows the exact clock cycle when the requested data will be ready. This allows the CPU to perform more instructions at a given time. Typical SDRAM transfers data at speeds up to 133 MHz
- Rambus DRAM (RDRAM) takes its name after the company that made it, Rambus. It was popular in the early 2000s and was mainly used for video game devices and graphics cards, with transfer, speeds up to 1 GHz.
- Double Data Rate SDRAM (DDR SDRAM) is a type of synchronous memory that nearly doubles the bandwidth of a single data rate (SDR) SDRAM running at the same clock frequency by employing a method called "double pumping," which allows the transfer of data on both the rising and falling edges of the clock signal without any increase in clock frequency.
- DDR1 SDRAM has been succeeded by DDR”, DDR3, and most recently, DDR4 SDRAM. Although operating on the same principles, the modules are not backward-compatible. Each generation delivers higher transfer rates and faster performance. The latest DDR4 modules, for example, feature fast transfer rates at 2133/2400/2666and even 3200 MT/s.

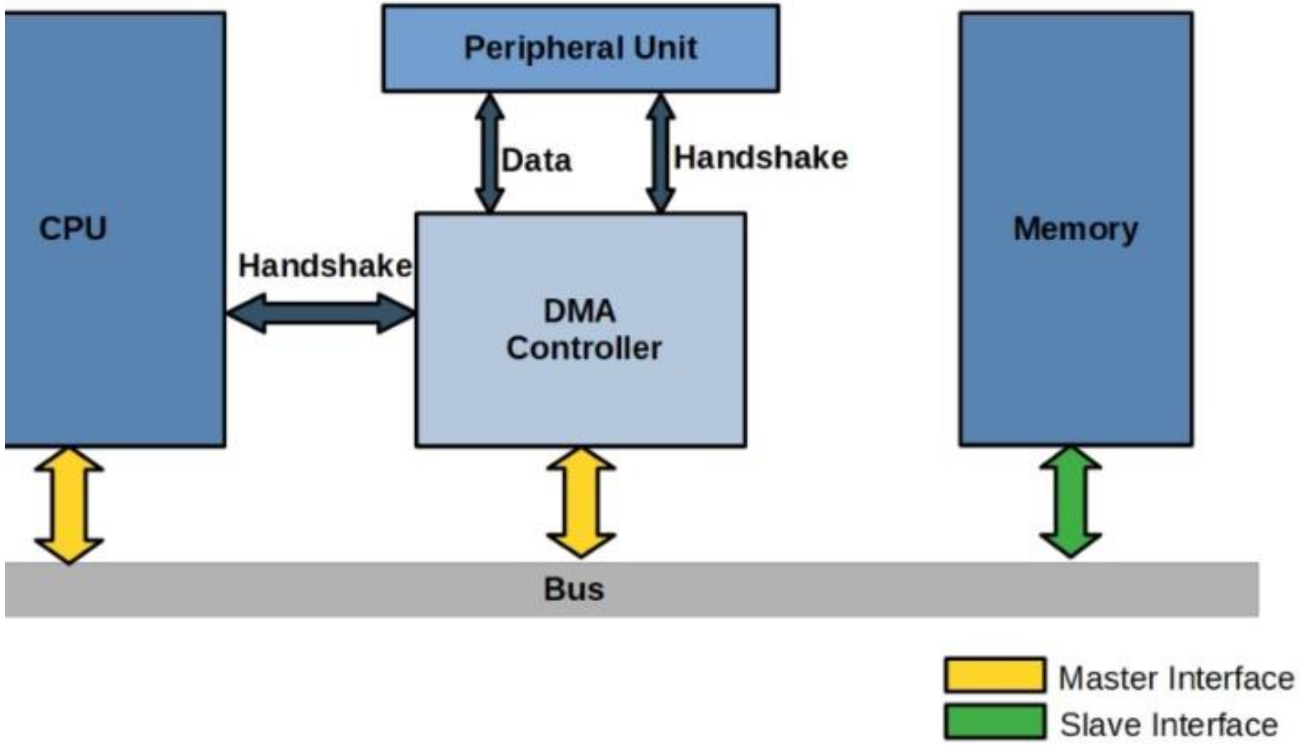

Computer Memory
Internal/Main/Primary
External/Secondary
ROM
(Non-Volatile)
RAM
(Volatile)
HDD
SSD
Compact
Disc
USB
Flash Drive
SRAM
DRAM
SDRAM
RDRAM
DDR
SDRAM
DDR1
DDR2
DDR3
DDR4

Direct Memory Access (DMA) in Embedded Systems

- Direct Memory Access (DMA) is a process of transferring data from one memory location to another without the direct involvement of the processor (CPU). The main benefit of using DMA is more efficient data movement in the embedded system.

Principle of Operation (DMA)

- There are many different types of DMA implementations, some of them for very specific use cases. In this article, we will focus on the general principles of operation. Let’s start with the simple system shown below in Fig.1.
- The functional unit that performs the operations for directly accessing the memory is called a **DMA controller**. On the simplified block diagram (Fig.1) we have a CPU, a RAM, a peripheral unit, and a DMA controller. All except the peripheral unit are connected on the same bus. As the CPU and the DMA controller must be able to initiate transfers they have master interfaces. Although the goal is to have DMA that operates independently, the CPU is the one that has to configure the DMA controller to perform transfers. The DMA controller can be dedicated to a specific DMA-capable peripheral unit (as shown in Fig. 1) or can be a more general DMA able to access various types of memory-mapped peripherals.



Fig,1 Simplified DMA block diagram

Why Use DMA?

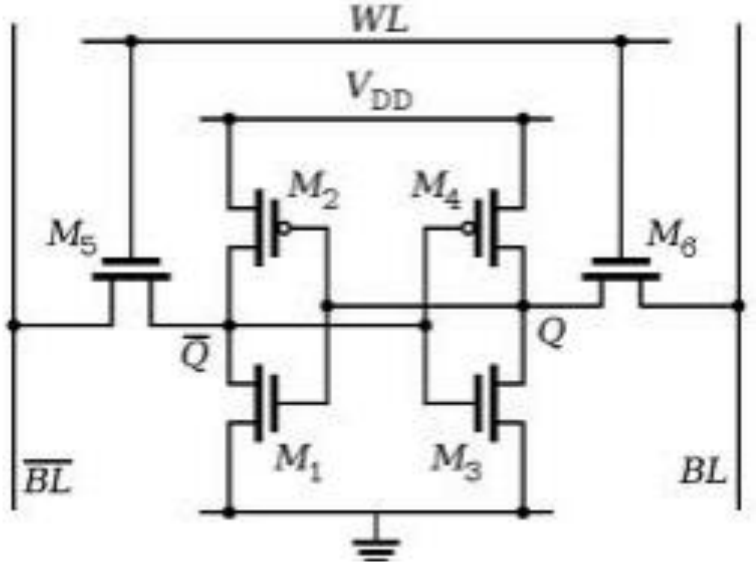
- DMA is used for moving data from one address of the memory to another. When used properly it can improve the efficiency of an embedded system. The CPU can be more focused on performing calculations, without having to waste too many instruction cycles for transferring data. This can result in improving the speed of our program.

- Another benefit can be the reduction of power consumption. The two common ways of transferring data without the use of DMA are:
  - based on interrupts– interrupt is generated when new data is available, and the CPU has to transfer it
  - polling – the CPU waits for new data to become available and then transfers it

- Both methods require the CPU to be awake. In contrast, some DMA controllers can perform data transfers while the CPU is in sleep mode (low-power mode).

Static RAM

- SRAMs are volatile
- Basic cell
  - Bistable core
    - 4T: uses pull-up resistors for M2 M4
    - 6T: uses P-FET for M2, M4
  - Access transistors
  - BL, BL are provided to improve noise margin
- 6T is typically used (but has poor density)
- Fast access times 0(10 ns)
- Read/write speeds are symmetric
- Read/write granularity is word

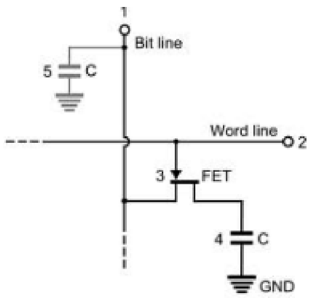


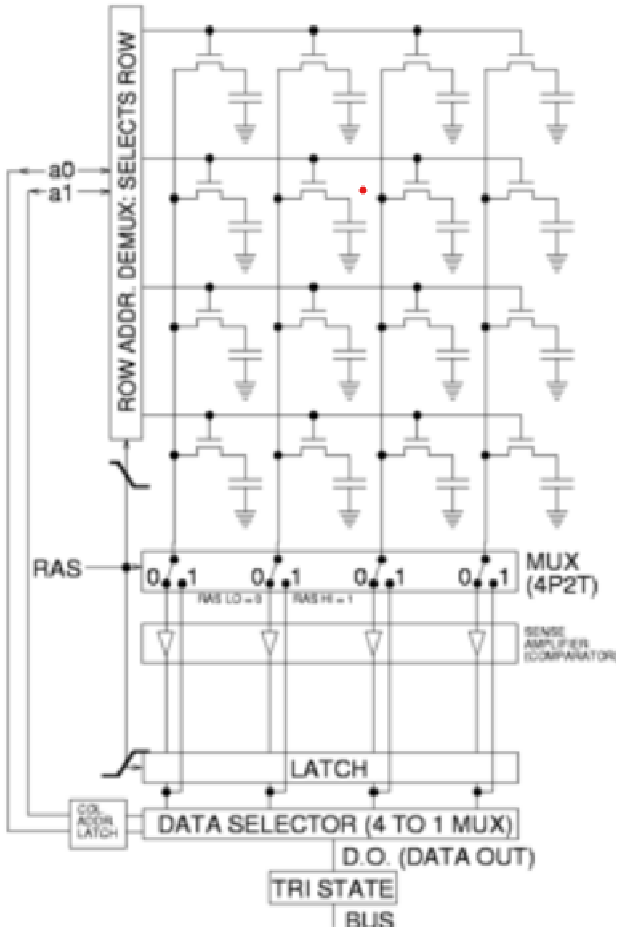
Dynamic RAM

- Requires only 1T and !c per cell
- Outstanding density and low cost
- Compare to the 6T‘s per SRAM cell
- Cost advantage to DRAM technology

- Small charges involved → relatively slow
  - Bit lines must be pre-charged to detect bits
  - Reads are destructive; internal writebacks needed

- Values must be refreshed periodically
  - Prevents charge from leaking away
  - Complicates control circuitry slightly





3: Single-board Computers

3.1 Beaglebone Black (BBB)

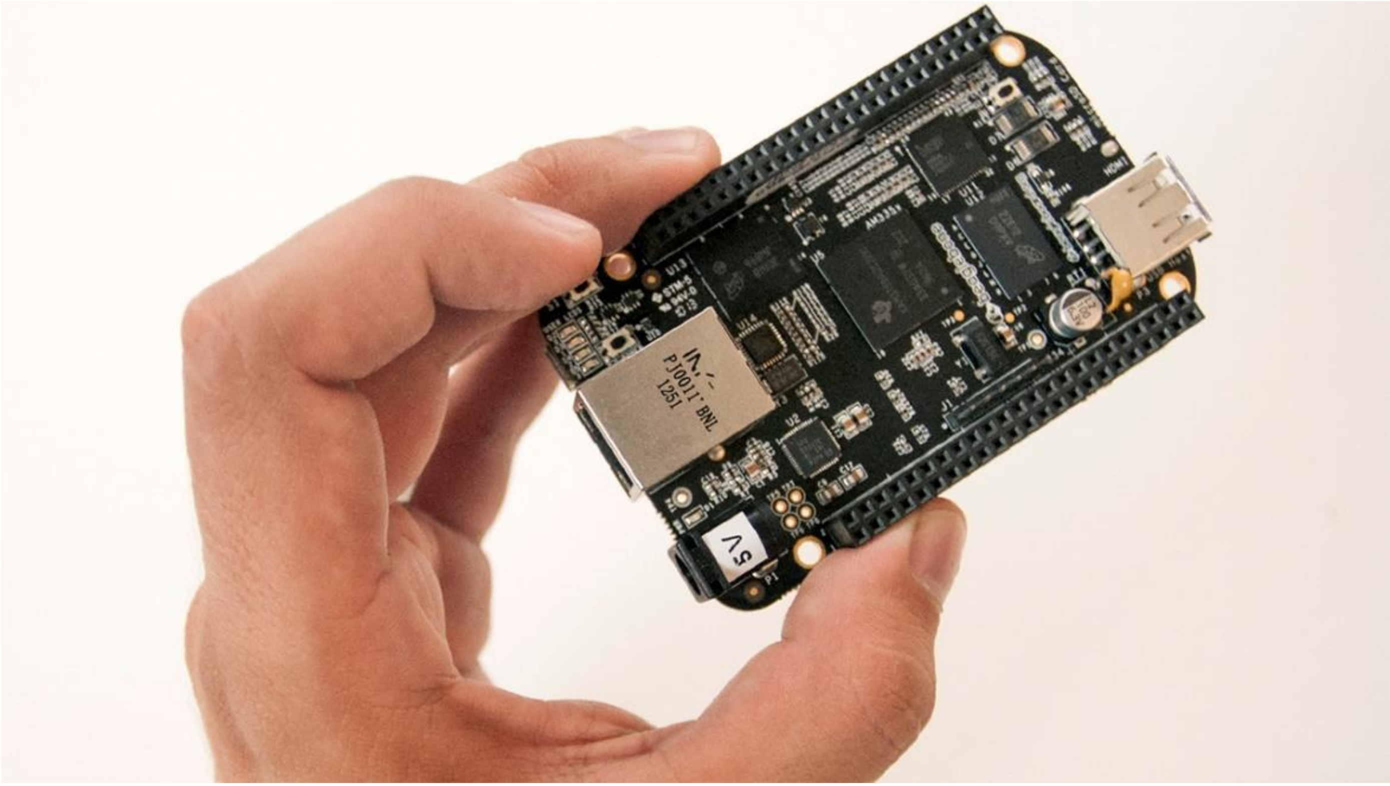
Introduction to the Beagle Bone Black

- Everybody has a smartphone smartphone... so what is the big deal behind a single board computer?
  - you can modify the hardware and software of this small and powerful device
  - adapt it to suit your own needs
  - create your own inventions
  - Price: $45–$55.

- Microprocessors
  - Memory
  - Processing speed
  - Wireless
- Beagle Bone Black
  - Various interfaces
  - Robotics control unit
  - Has its own flash memory
  - Suited for in and output

What is BeagleBone

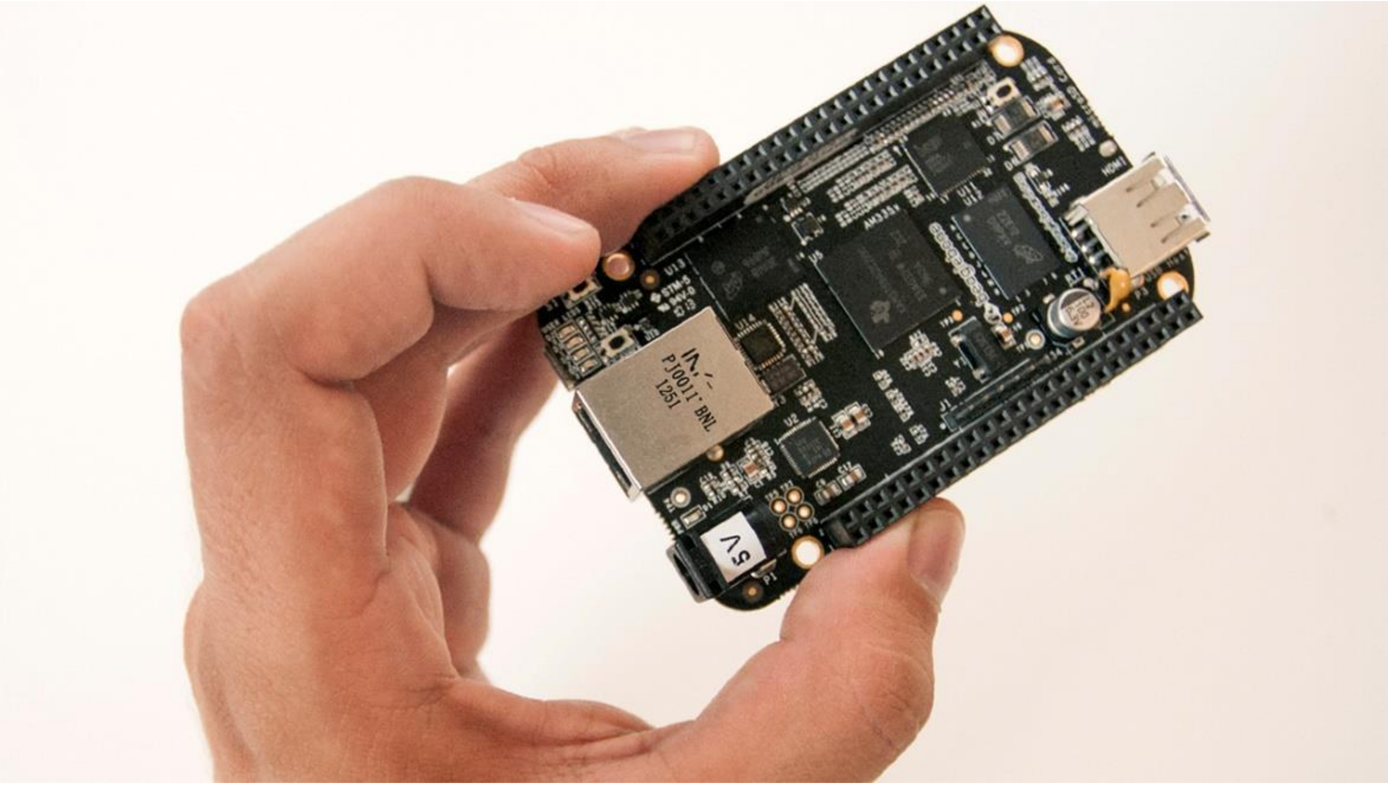
- Robotics - Embedded microprocessors
- Processor - Embedded in larger device
  - Microwave
  - Smartphone
  - Aeroplane
- Microprocessors
  - Memory
  - Processing speed
  - Wireless
- Beagle Bone Black
  - Various interfaces
  - Robotics control unit



Why use a BeagleBone?
Comparison Arduino, Rasp, BBB

Arduino:

- most basic
- highly extensible device
- utilizes low power and can run only one proc
- low processing speed.
- no GUI based IDE for programming
- advantage with Arduino is its flexibility to ext
- low cost of about 10 USD
- huge support base



Why use a BeagleBone?
Comparison Arduino, Rasp, BBB

Rasperry Pi:

•Raspberry Pi has programming IDE with GUI

• use for requirements that need high level of direct connectivity with the internet (databases)

BeagleBone Black :

• includes the power of Raspberry Pi and the flexibility of Arduino.

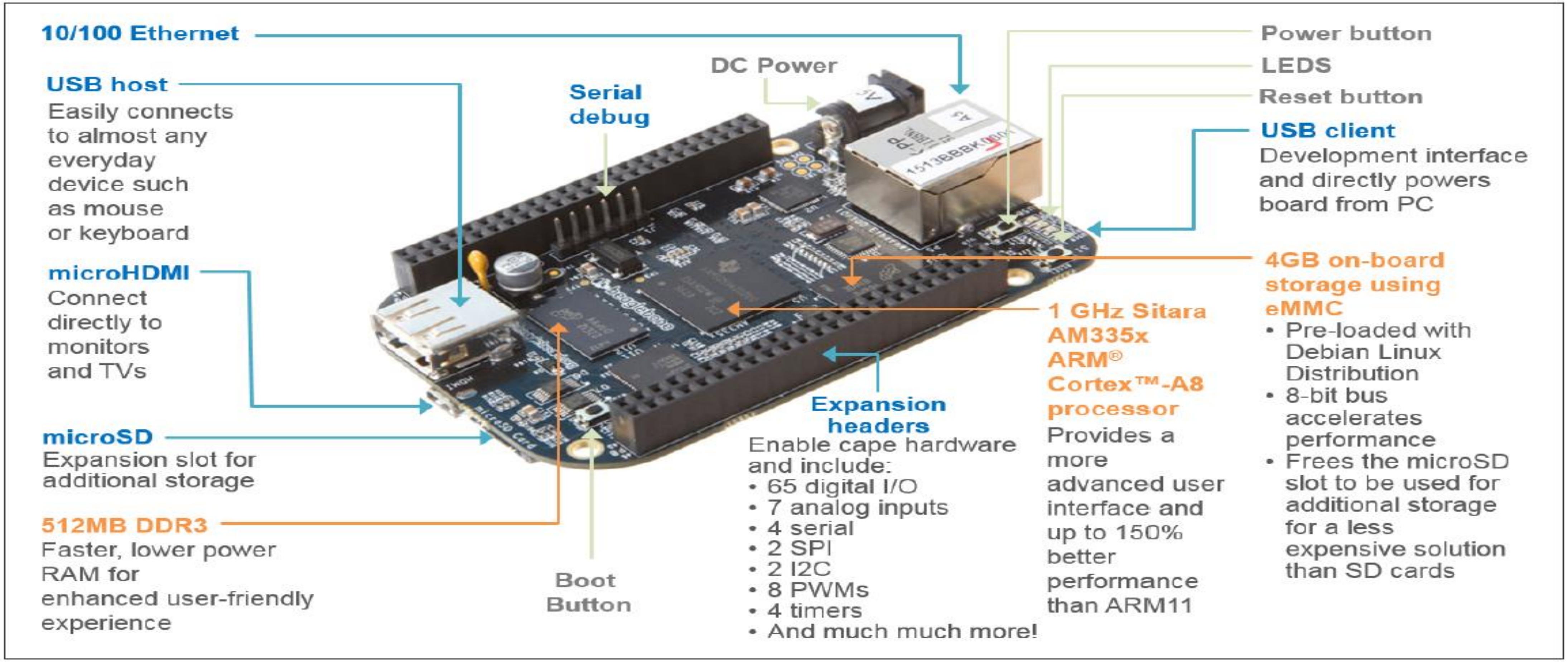
•It comes embedded with Linux OS and a memory.

• It has a pretty fast processing cycle and comes with lots of input and output pins for connecting external sensors and actuators. Its biggest limitation is that of its single USB port.

• Used for industrial and medical applications

• Limitation with HD video and real time

Beaglebone Black

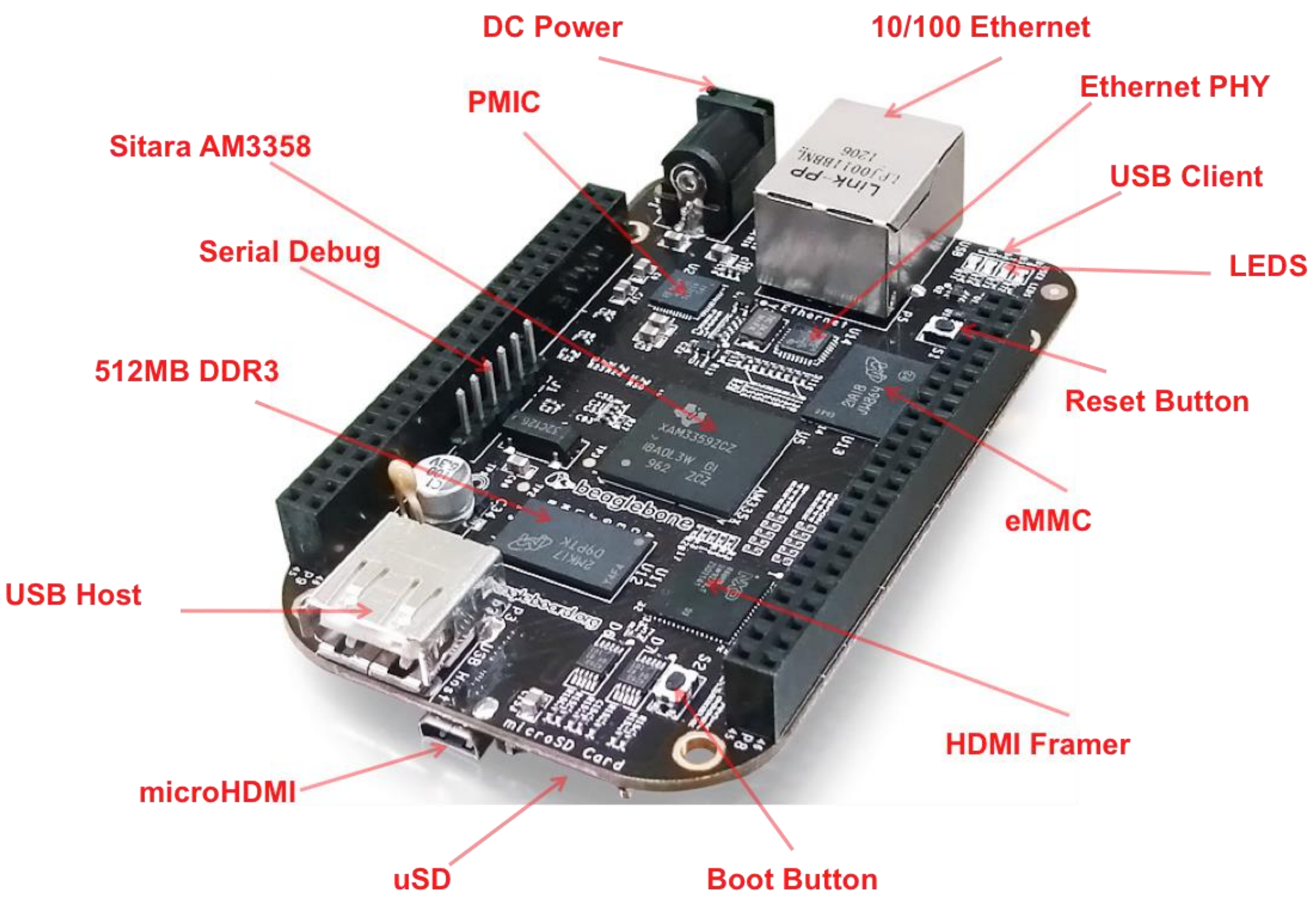


Main Features:

- Increased RAM to 512MB DDR3 RAM, the processor clock to 1 Ghz
- 4GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
- 2x PRU 32-bit microcontrollers
- HDMI
- Linux kernel 3.8
- BeagleBone Black flash memory enables it to ship with Debian GNU/Linux installed. Previous revisions shipped with Ångström Linux
- Perfect for in and output
- Affordable price

BeagleBoard

- Flash memory to support a fully capable OS
- Debian Linux
    - Networking
    - File system
    - Install and adapt everything
- Realtime
- Store many programms on your Beagle Bone
- Compile, run different programms


DC Power
10/100 Ethernet
PMIC
Ethernet PHY
Sitara AM3358
USB Client
Serial Debug
LEDS
512MB DDR3
Reset Button
eMMC
USB Host
HDMI Framer
microHDMI
uSD
Boot Button

Beaglebone Black Pinout Diagram

P9

Function	Physical Pins		Function
DGND	1	2	DGND
VDD 3.3 V	3	4	VDD 3.3 V
VDD 5V	5	6	VDD 5V
SYS 5V	7	8	SYS 5V
PWR_BUT	9	10	SYS_RESET
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPIO_CSO	17	18	SPIO_D1
I2C2_SCL	19	20	I2C_SDA
SPIO_DO	21	22	SPIO_SLCK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SP11_CSO
SP11_DO	29	30	GPIO_112
SP11_SCLK	31	32	VDD_ADC
AIN4	33	34	GND_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPWMO
DGND	43	44	DGND
DGND	45	46	DGND

LEGEND
Power, Ground, Reset
Digital Pins
PWM Output
1.8 Volt Analog Inputs
Shared I2C Bus
Reconfigurable Digital

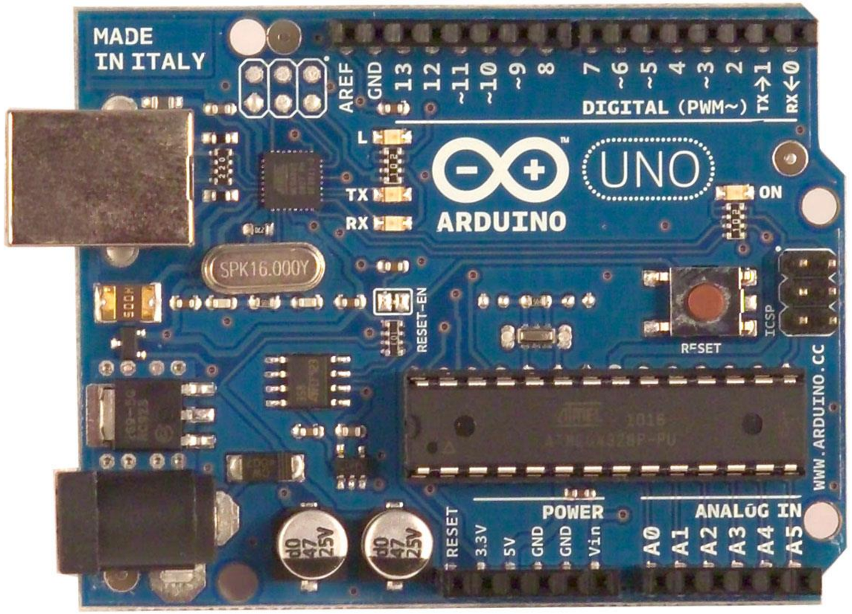
P8

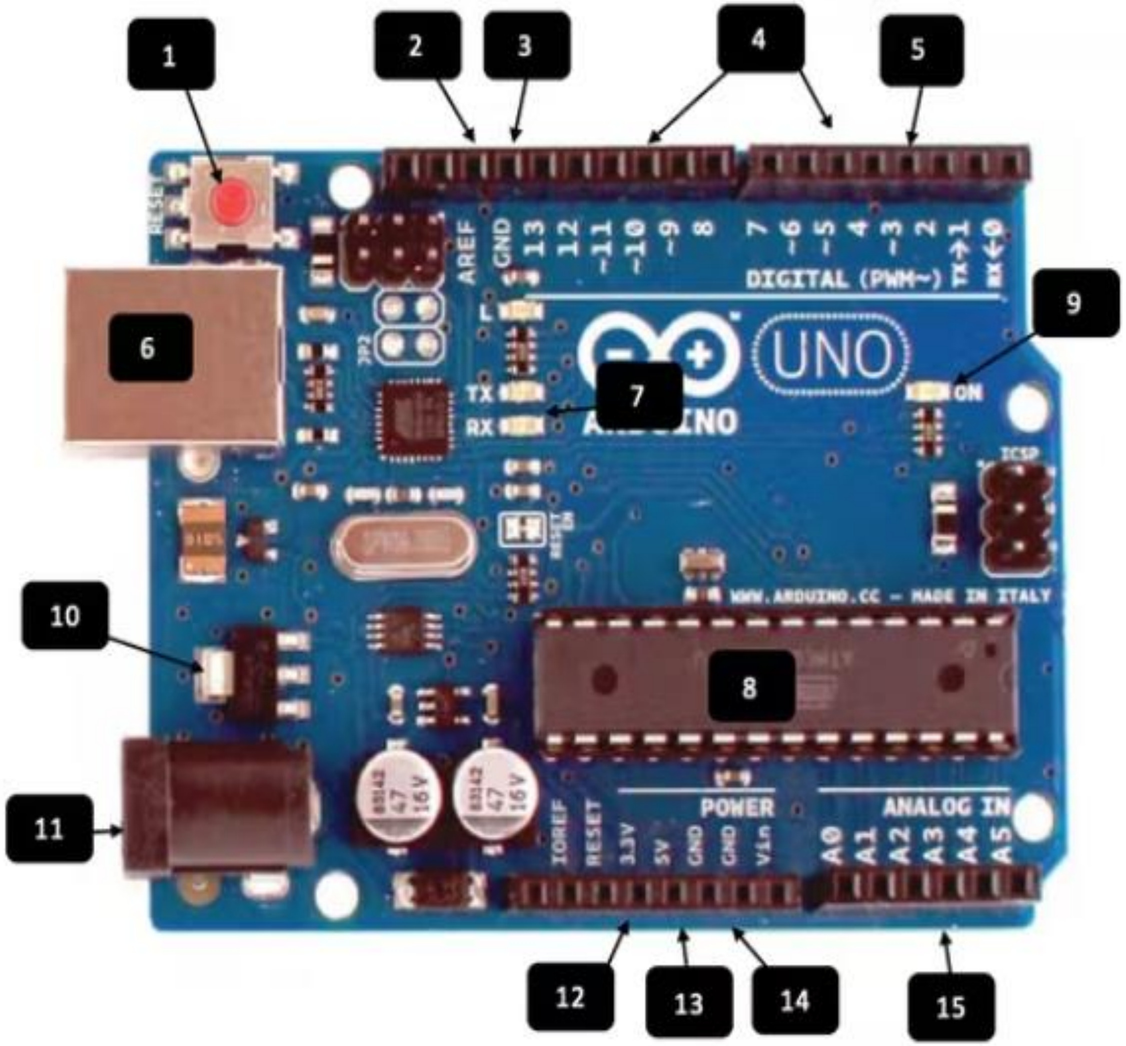
Function	Physical Pins		Function
DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DATO	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

3.2 Arduino

What is Arduino

- Arduino is an open-source platform based around programmable development boards that can be integrated into a range of simple and complex projects. The Arduino family consists of different types of development boards, with the most common being the Arduino UNO.
- An Arduino board contains a microcontroller that can be programmed to sense and control devices in the physical world. The microcontroller is able to interact with a large variety of components such as LEDs, motors, and displays. Because of its flexibility and sustainability, Arduino has become a popular prototyping development board that is widely used across the world.





1. Reset Button – This will restart any code that is loaded to the Arduino board
2. AREF – Stands for “Analog Reference” and is used to set an external reference voltage
3. Ground Pin – There are a few ground pins on the Arduino and they all work the same
4. Digital Input/Output – Pins 0-13 can be used for digital input or output
5. PWM – The pulse width modulation pins marked with the (~) symbol can simulate analog output
6. USB Connection – Used for powering up your Arduino and uploading sketches
7. TX/RX – LEDs to visualize data being transmitted and received from the board
8. ATmega328p – This is the microcontroller that stores the program and processes it
9. Power LED Indicator – This LED indicates the board is connected to a power source
10. Voltage Regulator – This controls the amount of voltage going into the Arduino board
11. DC Power Barrel Jack – This is used for powering your Arduino with a power supply
12. 3.3V Pin – This pin supplies 3.3 volts of power to your projects
13. 5V Pin – This pin supplies 5 volts of power to your projects
14. Ground Pins – There are a few ground pins on the Arduino and they all work the same
15. Analog Pins – These pins can read the signal from an analog sensor and convert it to digital

What is Arduino used for?

- Arduino has many uses from coding simple projects to complex systems and demonstrating real-life applications. The Arduino IDE is used to write programs and compile them to be loaded on the board. This supports a common programming language, C++.
- Throughout the rest of the course, we will be exploring the Arduino board and learning programming techniques to build systems.

3.3 Raspberry Pi

What is Raspberry Pi

- The Raspberry Pi is a low-cost, credit-card-sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python.

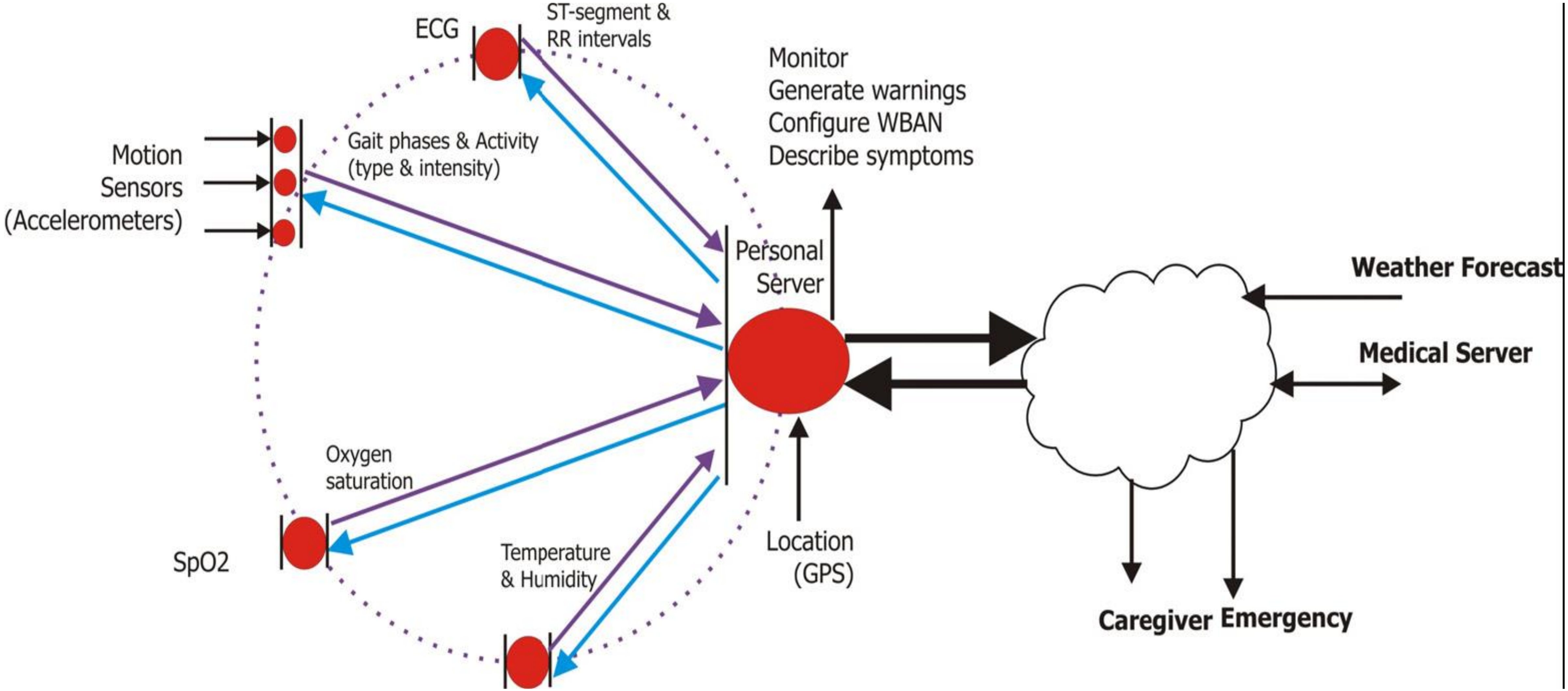
What to make with Raspberry Pi

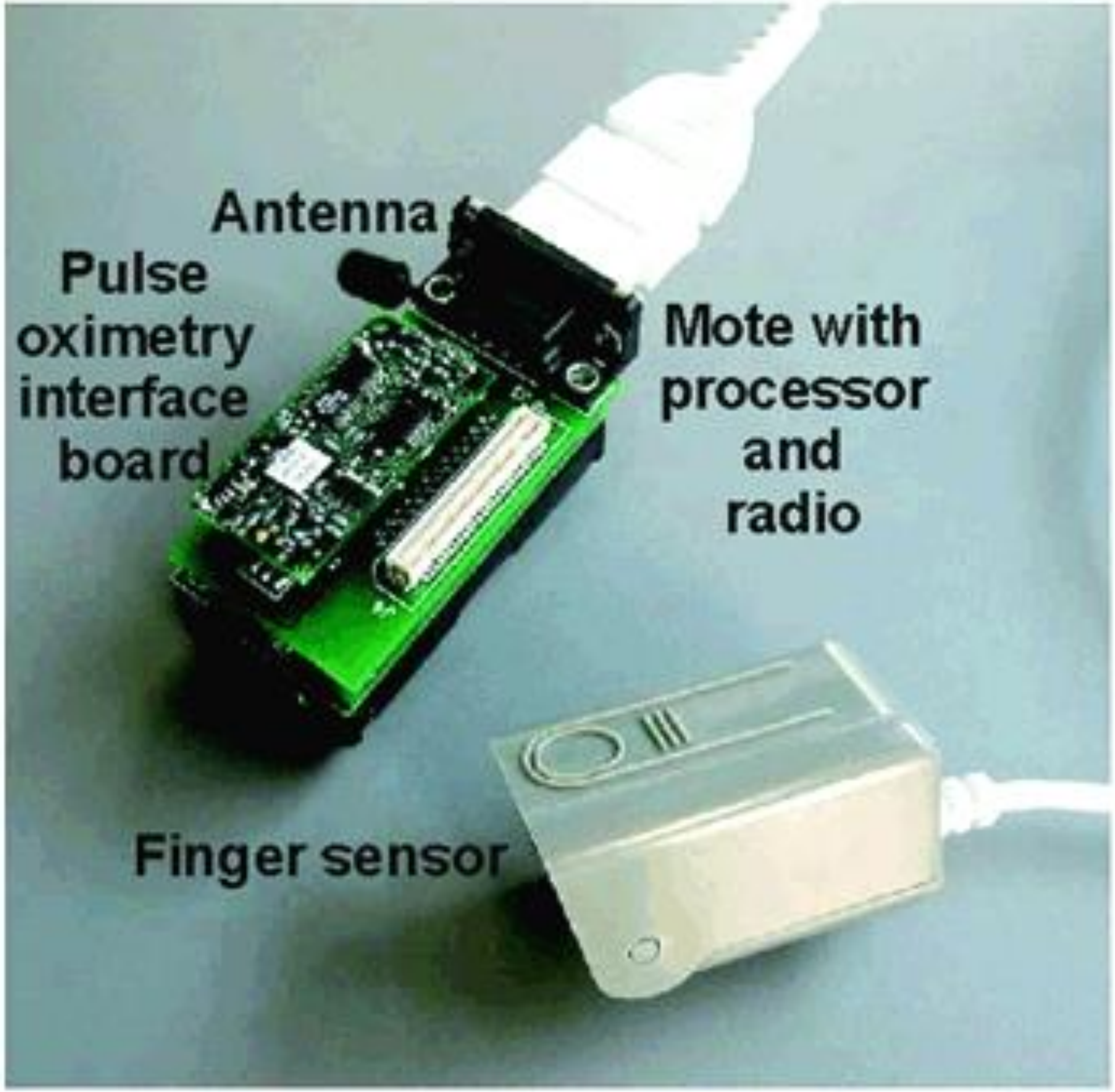
- A Single board computer(such as Raspberry Pi) is a computer that contains basic units i.e. ram memory, input-output unit, and microprocessor but unlike normal computers, it is not possible to expand the hardware features. For example, it does not contain a SLOT to increase the RAM memory from 1GB to 2GB.
- Since Raspberry Pi is designed as a single board and does not have a structure open to development for extra hardware to be installed on it, its cost is quite low. Single-board computers are not used as personal computers but are used in engineering projects where heavy computing is required i.e. robotics, IoT, image processing, etc.

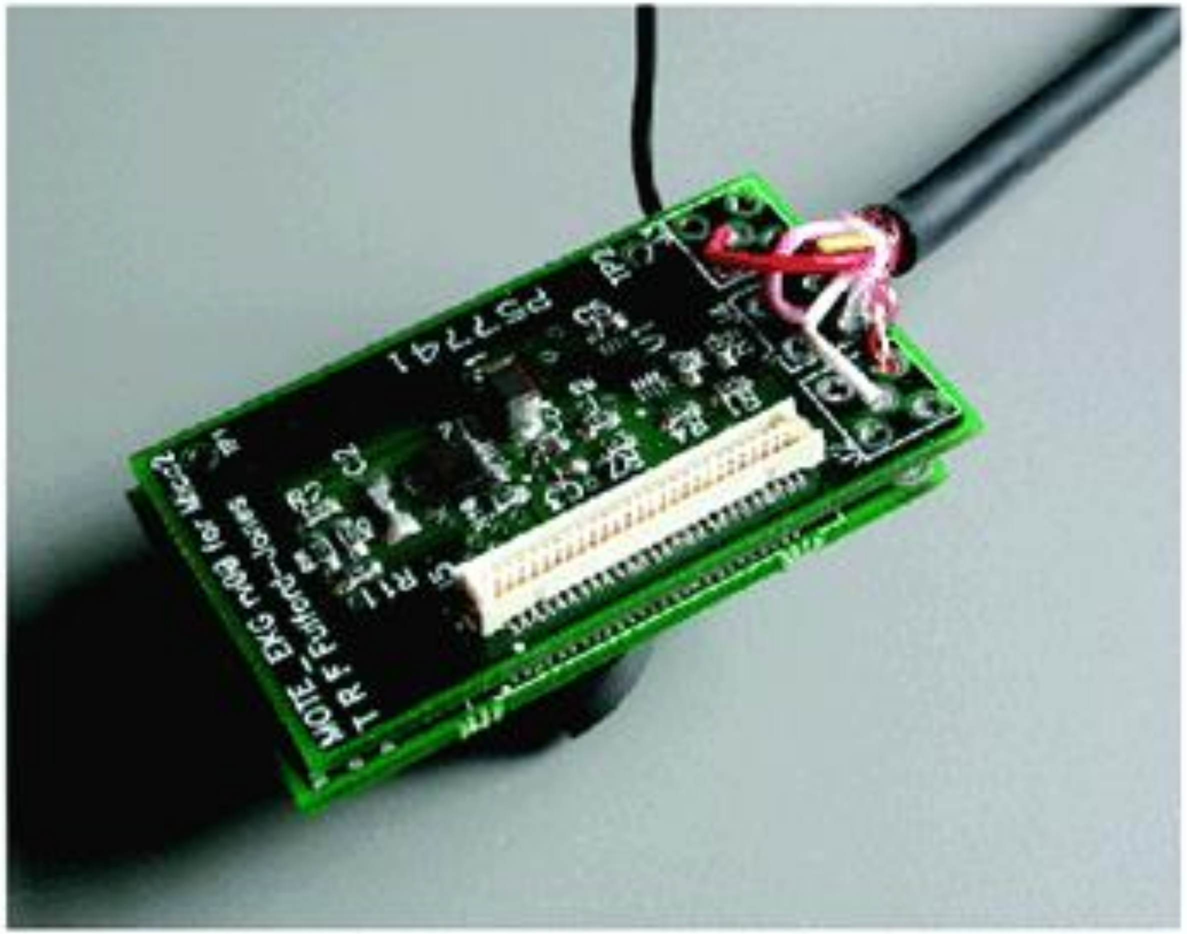
Where is the Raspberry PI used ?

- Replace your desktop PC with a Raspberry Pi
- Media Center
- Retro gaming machine
- server
- Control a robot
- Build a stop motion camera
- security system
- A home automation system with Arduino
- Learn how to code(Python-c++-code blocks-)
- Build a smart mirror
- Bluetooth and Wifi Projects
- AI Projects

4: Applications of embedded systems


ECG
ST-segment &
RR intervals
Monitor
Generate warnings
Configure WBAN
Describe symptoms
Motion
Sensors
(Accelerometers)
Gait phases & Activity
(type & intensity)
Personal
Server
Weather Forecast
Medical Server
Oxygen
saturation
SpO2
Temperature
& Humidity
Location
(GPS)
Caregiver
Emergency


Antenna
Pulse oximetry interface board
Mote with processor and radio
Finger sensor



E-Health Applications: State of the Art

Problems right now are:

- How to handle the big data amount
- Detecting the right signals
- Interfacing with other devices (servers, mobile phones, different OS and devices)

Wearable systems have been proposed:

- Integrated wireless transmission
- GPS (Global Positioning System) sensors, and local processing

Need for

- Processing power (more than mobile phones)
- Small computers which are adaptable (Linux as the preferred OS)
- In and output for multimedia (images, signal,s and video) → HDMI (High Definition
- Multimedia Interface)
- Getting real-time (fast processing, parallel computing)

Digital/Analog

- Analog and digital signals are used to transmit information, usually through electric signals. In both these technologies, the information, such as any audio or video, is transformed into electric signals.
- The difference between analog and digital technologies is that in analog technology, information is translated into electric pulses of varying amplitude. In digital technology, the translation of information is into binary format (zero or one) where each bit is representative of two distinct amplitudes.

What is analog technology?

- People accept digital things easily enough, often by thinking of them as computerized, and perhaps not even worth trying to understand. But the concept of analog technology often seems more baffling—especially when people try to explain it in pages like this. So what's it all about?

What is digital technology?

- Digital is entirely different. Instead of storing words, pictures, and sounds as representations on things like plastic film or magnetic tape, we first convert the information into numbers (digits) and display or store the numbers instead.

Which is better to be used?

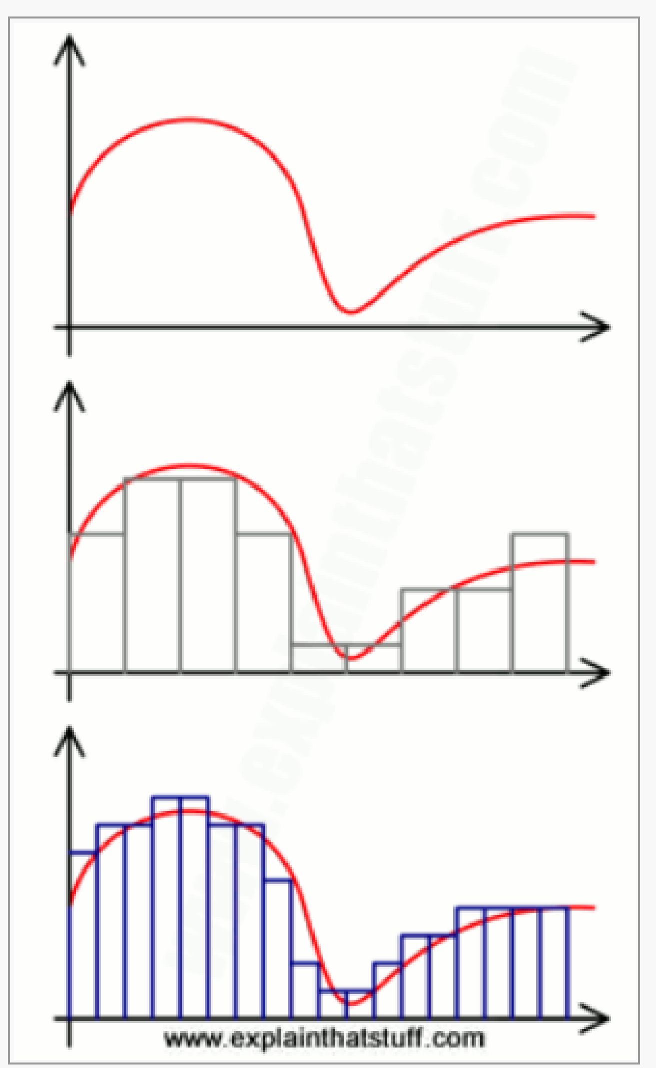
- Just because digital technology has advantages, that doesn't mean it's always better than analog.
- Surprisingly, analog watches can also keep time better than quartz ones: the day-to-day variations in a mechanical, analog watch tend to cancel one another out, while those in an electronic quartz watch tend to compound one another.
- Museums still have paper documents (and ones written on clay or stone) that are thousands of years old, but no one has the first email or cellphone conversation.
- That's why, though the future may be digital, analog technology will always have its place!

What is sampling?

- The process of converting analog to digital is called analog-to-digital conversion (ADC) or, more informally, sampling.
- Sampling simply means "measuring at regular intervals"—and it's easiest to understand with an example.

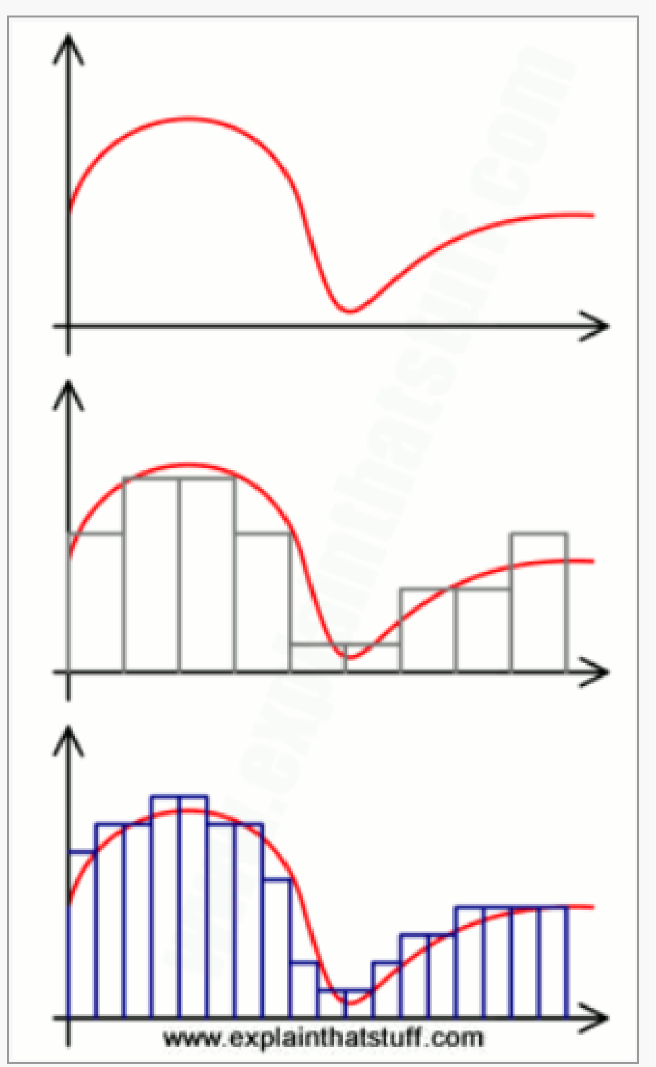
Example sampling

- Let's suppose I'm talking to you on my cellphone. The sound of my voice is really waves of energy that travel through the air to the phone's microphone, which converts them into electrical signals. The sound waves and the signals are both continuously varying waveforms—they're analog information— and they look like the upper graph in the diagram.
- A cellphone transmits sound in digital form, so those analog waves need to be converted into numbers. A circuit inside the phone measures the size of the waves many times each second and stores each measurement as a number.
- You can see in the middle figure that I've turned the first graph into a very approximate bar chart.
- If each bar represents one second of time, we can represent this chart by nine numbers (one number for the height of each bar): 5-7-7-5-1-1-3-3-5.
- We could send those numbers through the air as radio waves to another phone, which would run the process in reverse.



DRAWBACK

- Some information is going to get lost in the process of converting the sound to digital and back again because the measurement I've made doesn't precisely capture the shape of the original wave: it's only a crude approximation. What can I do about this?
- I could make *more measurements*, by measuring the sound wave twice as often. That means doubling what's called the sampling rate. Now, as you can see in the bottom chart, I get twice as many measurements and my sound wave is represented by eighteen numbers: 6-7-7-8-8-7-7-5-2-1-1-2-3-3-4-4-4-4.
- The more I increase the sampling rate, the more accurate my digital representation of the sound becomes—but the more digital information I create and the more space I need to store it.



Sampling rate and bit rate

- Broadly speaking, the bit rate is the amount of information captured each time the music is sampled. So a higher bit rate means more information is captured and the analog information is turned into digital information more accurately. Higher-quality music tracks may have a higher bit rate, but the tracks will take up far more space on your computer and take longer to download.

Definitions of Analog vs. Digital signals

- An Analog signal is any continuous signal for which the time varying feature (variable) of the signal is a representation of some other time varying quantity, i.e., analogous to another time varying signal. It differs from a digital signal in terms of small fluctuations in the signal which are meaningful.
- A digital signal uses discrete (discontinuous) values. By contrast, non-digital (or analog) systems use a continuous range of values to represent information. Although digital representations are discrete, the information represented can be either discrete, such as numbers or letters, or continuous, such as sounds, images, and other measurements of continuous systems.

Differences in Usage in Equipment

- Many devices come with built in translation facilities from analog to digital. Microphones and speaker are perfect examples of analog devices. Analog technology is cheaper but there is a limitation of size of data that can be transmitted at a given time.
- Digital technology has revolutionized the way most of the equipments work. Data is converted into binary code and then reassembled back into original form at reception point. Since these can be easily manipulated, it offers a wider range of options. Digital equipment is more expensive than analog equipment.

Comparison of Analog vs Digital Quality

- Digital devices translate and reassemble data and in the process are more prone to loss of quality as compared to analog devices. Computer advancement has enabled use of error detection and error correction techniques to remove disturbances artificially from digital signals and improve quality.

Differences in Applications

- Digital technology has been the most efficient in the cellular phone industry. Analog phones have become redundant even though sound clarity and quality were good.
- Analog technology comprises natural signals like human speech. With digital technology, this human speech can be saved and stored in a computer. Thus digital technology opens up the horizon for endless possible uses.

Digital I/O vs Analog I/O

- The automation technology around us involves both digital I/O and analog I/O. They are commonly used for microcontrollers and machine learning such as Programmable Logic Controller (PLC) and Distributed Control System (DCS). The PLC receives and processes input data and produces its output depending on the program line.
- Both Digital and Analog I/O are fundamental for signal processing circuits and applications. Both of them have their own characteristics, advantages, and disadvantages. Now we will learn about digital I/O and analog I/O, everything about their differences, and also answers the frequently asked questions.
- Both analog I/O and digital I/O are mainly used in the industrial sector, power industry, and of course robotic technology. A Programmable Logic Controller or PLC is a simple example of this I/O.

What is I/O – Input Output

- We can't start learning about digital I/O and analog I/O if we haven't known about what is I/O. Maybe some of you have known that I/O stands for Input/Output. The input will determine how machines should act according to the program line inside its controller. The program will determine what output should be performed by the machine.
- Input is the energy or information that enters a system while output is the energy or information that leaves a system. These input and output things are the methods we can use to interact with the machine and vice versa. Input is usually related to switches, potentiometers, sensors, cameras, and many more. Output is usually related to electric motors, lighting devices, alarms, and many more.

Differences Analog I/O vs Digital I/O

- We can differentiate between analog I/O and digital I/O by their sensors and signal processing. As an illustration, we can use water level control. This water level control will tell us whether the water inside the water tank is full or not. Of course, using digital I/O or analog I/O will give us different results. Keep in mind that digital data only has values "0" and "1". Meanwhile, analog data can range between 0 and 1. We can read about it below

Analog I/O vs Digital I/O: Data Processing

- Analog data is processed continuously. Every time the input changes, the output will also change. There is almost no delay between every change of the input. The delay depends on the electron's movement speed which we already know that it is almost impossible to know about its exact time.
- Digital data is processed in discrete time delay. The delay depends on the time delay listed on the device's datasheet. The time delay is also calculated from sampling frequency, depending on the crystal or clock component inside the device.

Why Digital I/O is better

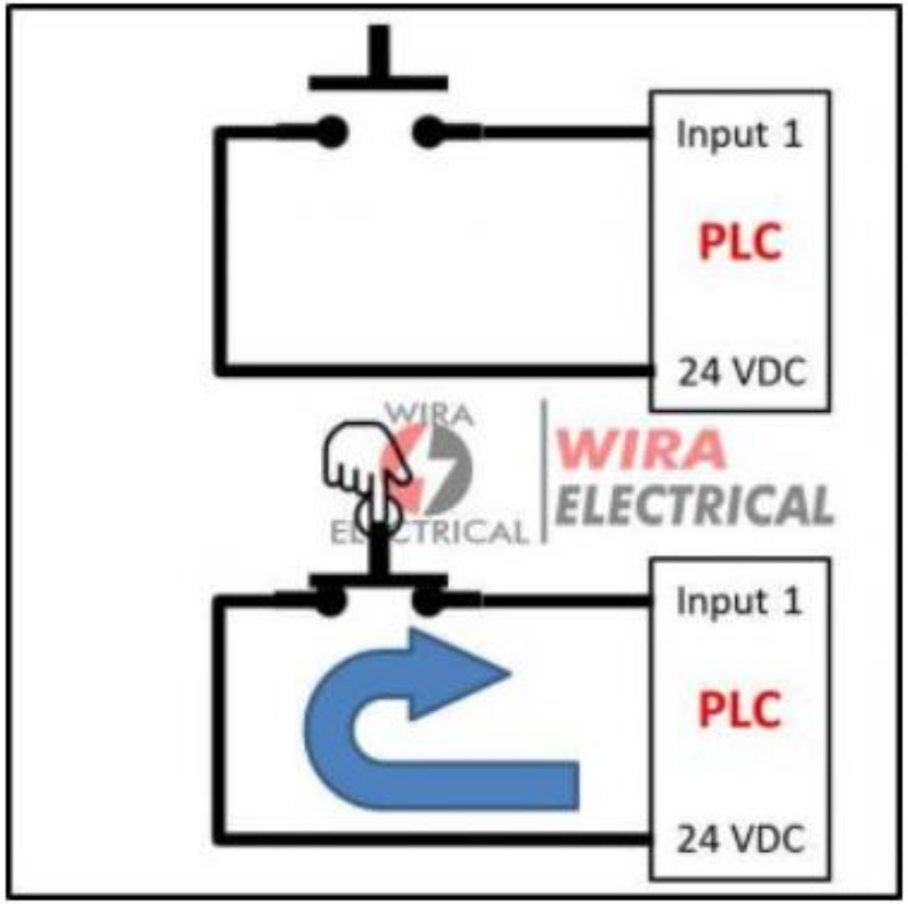
- When it comes to advanced technology such as robotics, digital input, digital output, and digital processing is superior to analog.

1. Since we have to program the robot, we need to use a computer. And as you have known, computers use digital data processing.
2. Digital data has less electrical noise when transmitted.
3. Digital data processing costs less than analog data processing and has a wide variety of applications

- Aside from digital I/O superiority in advanced technologies, our world has a majority of analog I/O and will always be like that. There is a trick to reducing electrical noise by analog data, we need to convert it to digital data as soon as possible. This method can be achieved by Analog to Digital Converter (ADC).

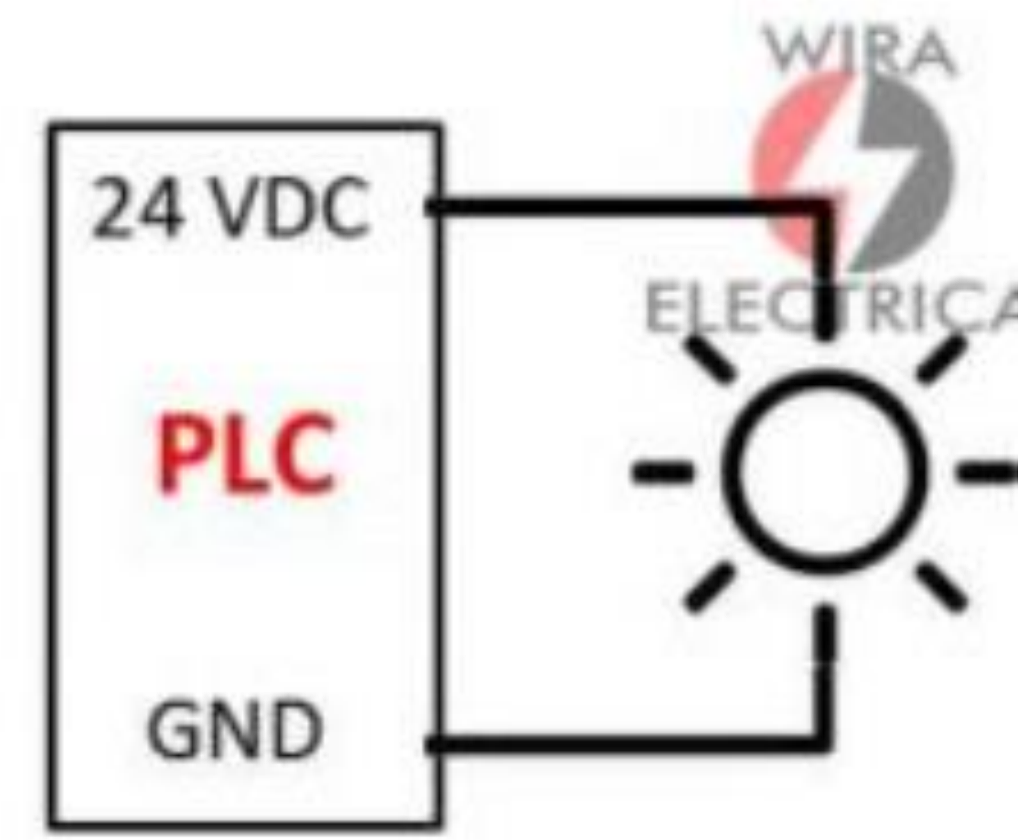
What is Digital Input

- Digital input used digital data which are binary numbers “0” and “1”. These “0” and “1” are basically voltage values. The “0” represents zero volts or no voltage input. The “1” represents 5V to 220V or there is voltage input. To achieve this, we can use sensors, switches, or push buttons to connect or disconnect the voltage source’s path. Since we only have “0” and “1”, it means they can only be used for “ON and “OFF”.



What is Digital Output

- Like digital input, we only use binary data (0 and 1) to produce output. The simplest example of digital output is a light bulb. The output will turn on the light bulb or turn off the light bulb. Other examples can be found on contacts of a relay, solenoid valve, lamp, or another example which is able to operate on ON or OFF conditions.



What is Analog Input

- Unlike digital input, the analog input can receive continuous change depending on the electron’s speed (it’s almost instant). Of course, unlike the digital input which uses constant or fixed voltage as their data, analog input uses various voltages between 0-100% depending on the maximum voltage. Let’s say we have 10VDC input so the analog input can range between 0-10VDC or 0-100% of 10VDC.

- For example, we can proportionally use this range with a potentiometer. We use 10VDC as input connected with a potentiometer. When the potentiometer reaches 50% resistance, it will produce 5VDC (50% from maximum VDC). If the resistance reaches 90%, the voltage will be 9VDC (90% from maximum VDC). Its scaling value is free for us to determine.

What is Analog Output

- Just like analog input, the analog output also produces continuous results. If the input is changed, the output is also changed almost instantaneously. The implementation of analog output can be seen from the meters such as pressure meter, water level meter, etc.

What is the difference between Digital IO and Analog IO?

- Analog I/O uses continuous signal and process it almost instantly when there is a change on the input. Digital I/O uses discrete signal and process time depending on clock speed or frequency.

What is digital output and analog output?

- Digital output means the outputs are “0” or “1”, “ON” or “OFF”. Analog output means the outputs are time-varying signal like sinusoidal wave.

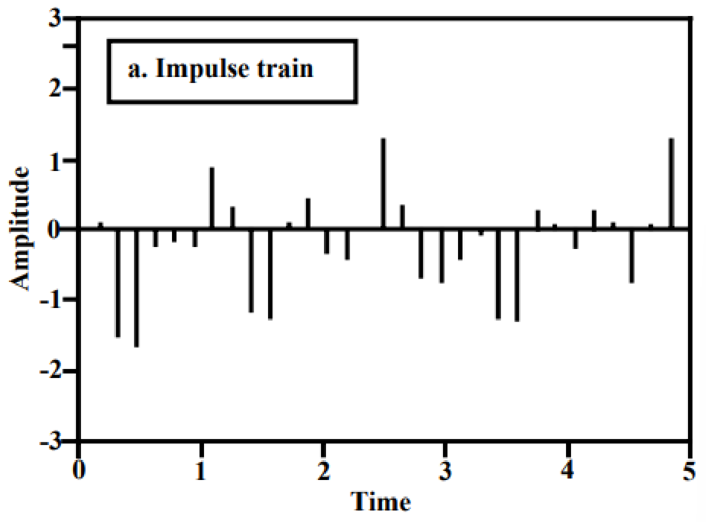


Fig. 18.4(a) The analog equivalent of digital words

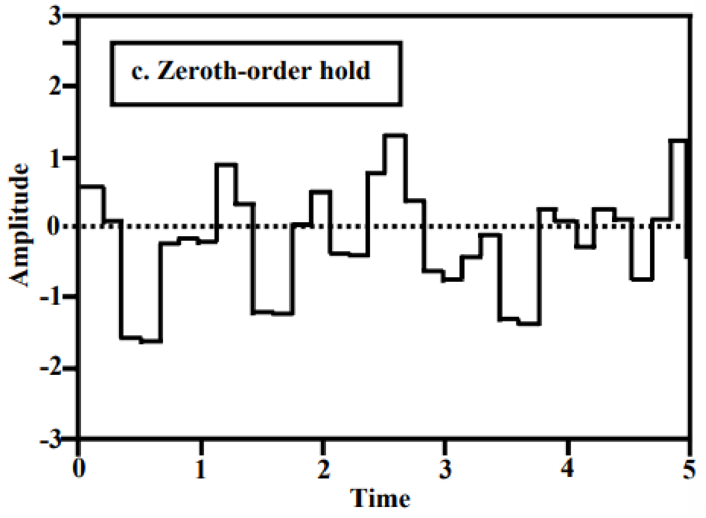


Fig. 18.4(b) The analog voltage after zero-order hold

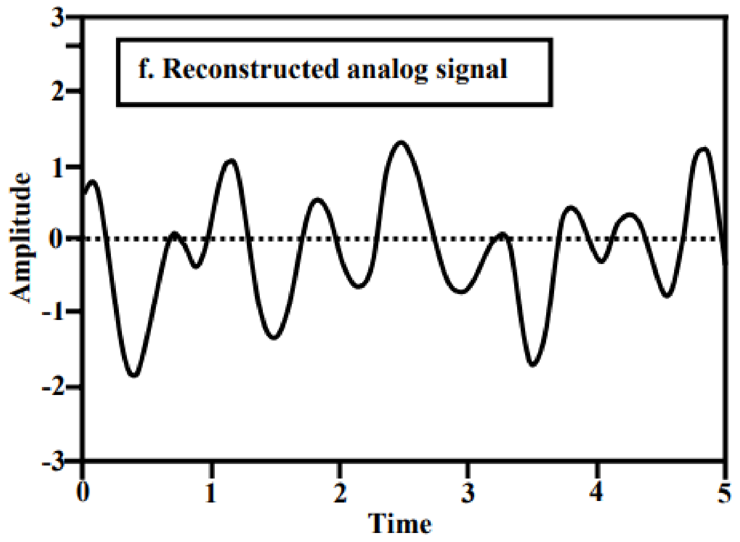
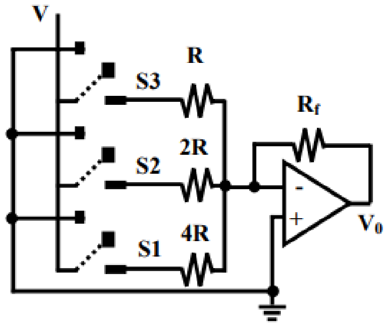


Fig. 18.4(c) The reconstructed analog signal after filtering

The DA Converter

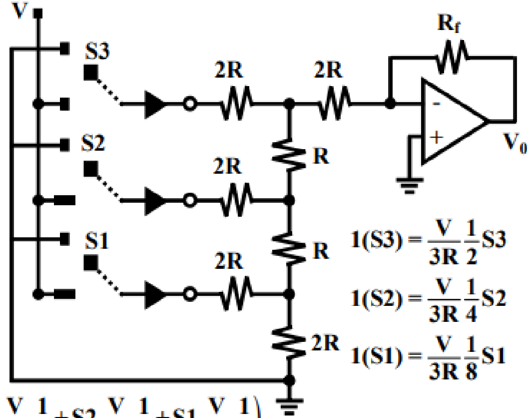
- Fig. 18.5(b) depicts the R-2R ladder network.
  The disadvantage of the binary-weighted register is the availability and manufacturing of exact values of the resistances.
  Here also the output is proportional to the binary-coded decimal number.
- The output of the above circuits is given in
  Fig. 18.5(a) and 18.5(b) are equivalent to analog values as shown in Fig. 18.4(a). However, to reconstruct the original signal this is further passed through a zero-order hold (ZOH) circuit followed by a filter (Fig.18.2).
  The reconstructed waveforms are shown in Fig. 18.4(b) and 18.4(c).



	S3	S2	S1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$$V0 = -R_f\left(S3\frac{V}{R} + S2\frac{V}{2R} + S1\frac{V}{4R}\right)$$
$$= \frac{-R_f}{4R}V(4S3 + 2S2 + S1)$$

Fig. 18.5(a) The binary weighted register method



$$I(S3) = \frac{V}{3R}\frac{1}{2}S3$$
$$I(S2) = \frac{V}{3R}\frac{1}{4}S2$$
$$I(S1) = \frac{V}{3R}\frac{1}{8}S1$$

$$V0 = -R_f\left(S3\frac{V}{3R}\frac{1}{2} + S2\frac{V}{3R}\frac{1}{4} + S1\frac{V}{3R}\frac{1}{8}\right)$$
$$= -R_f\frac{V}{24R}(4S3 + 2S2 + S1)$$

Fig. 18.5(b) R-2R ladder D-A conversion circuit

The AD Converter

- The ADC consists of:
  - Sampler
  - Quantizer
  - Coder

  - ARIAN: zu wenig info, keine grafik!!

The Sampler

- The sampler in the simplest form is a semiconductor switch as shown below. I is followed by a hold circuit which a capacitor with a very low leakage path.

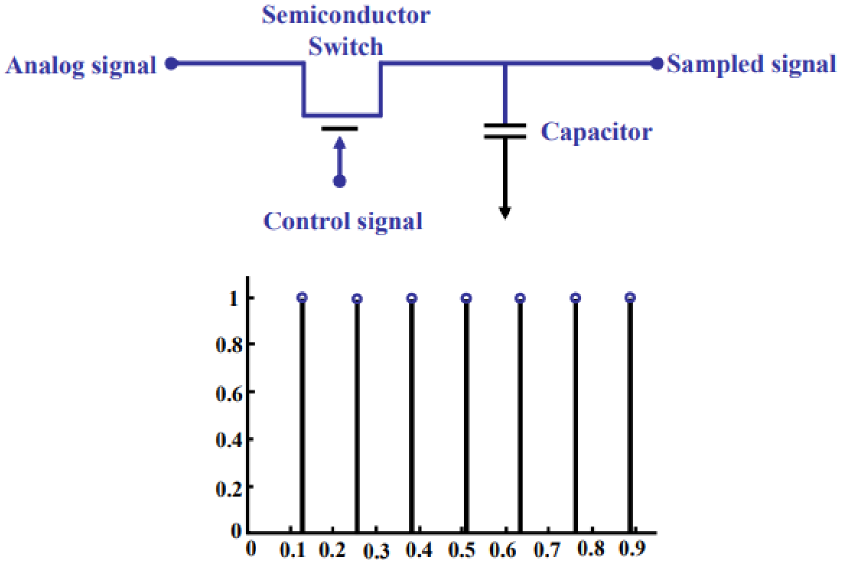


Fig. 18.6 The Sample and Hold Circuit

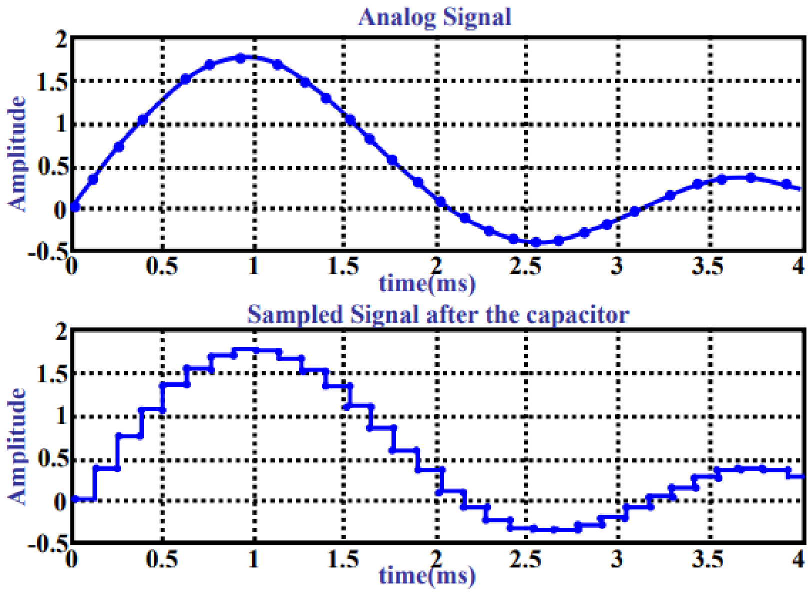


Fig. 18.7 Sample and Hold Signals

The Quantizer

- The hold circuit tries to maintain a constant voltage till the next switching. The quantizer is responsible to convert this voltage to a binary number. The number of bits in a binary number decides the approximation and accuracy. The sample hand hold output can assume any real number in a given range. However because of the finite number of bits (say N) the levels possible in the digital domain are 0 to 2N-1 which corresponds to a voltage range of 0 to V volts

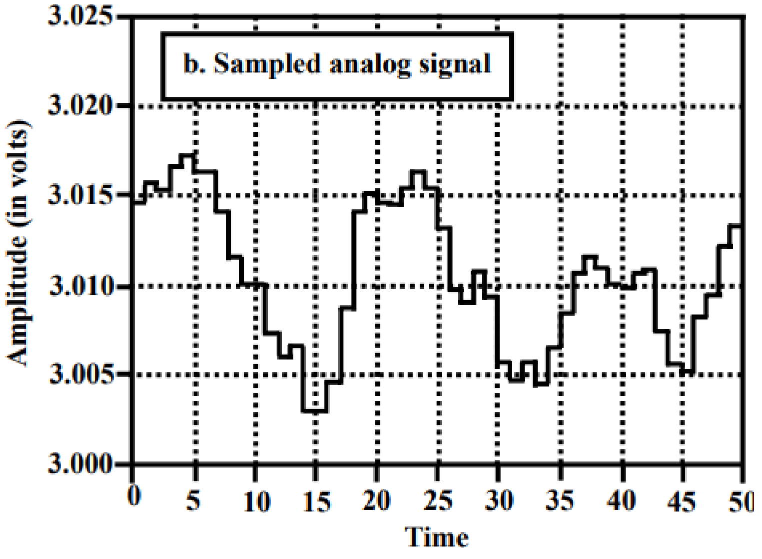


Fig. 18.8(a) Hold Circuit Output

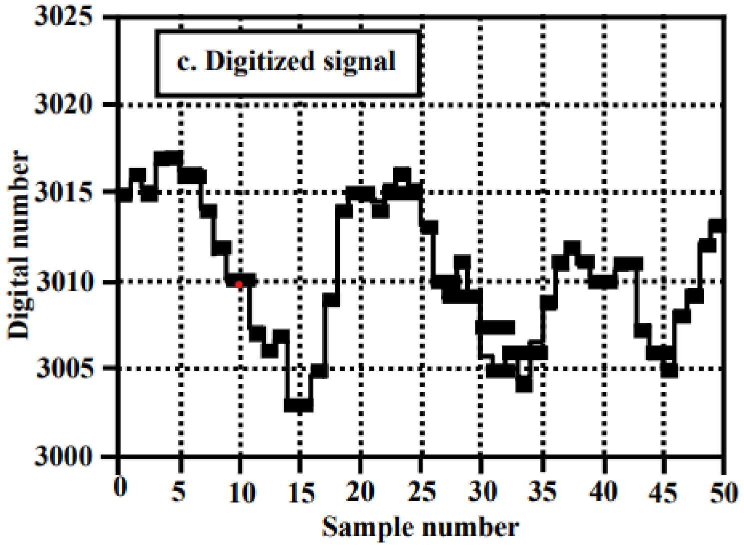
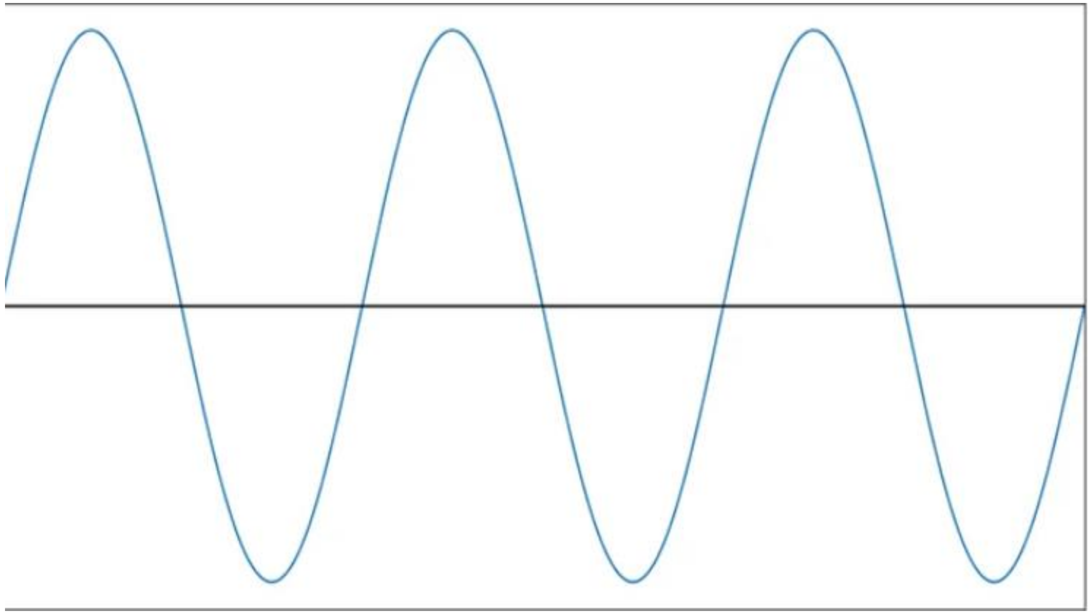


Fig. 18.8(b) The Quantized Value

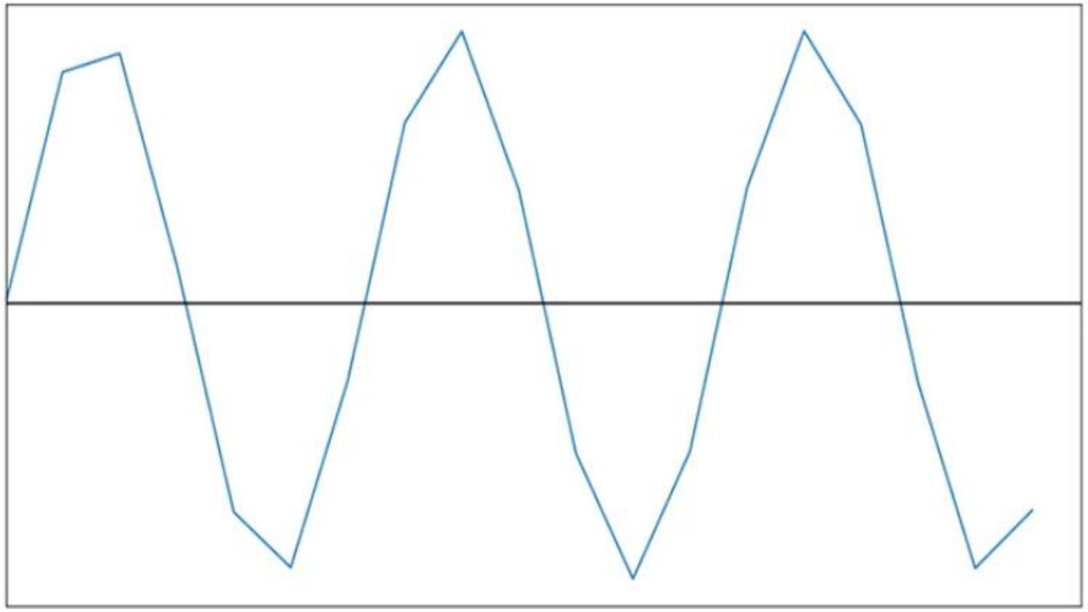
The Sampling Theorem:

- is a fundamental theoretical principle that governs the design of mixed-signal electronic systems.
- Modern technology as we know it would not exist without analog-to-digital conversion and digital-to-analog conversion. In fact, these operations have become so commonplace that it sounds like a truism to say that an analog signal can be converted to digital and back to analog without any significant loss of information.

Start a signal that looks like this:



And digitize it into this:

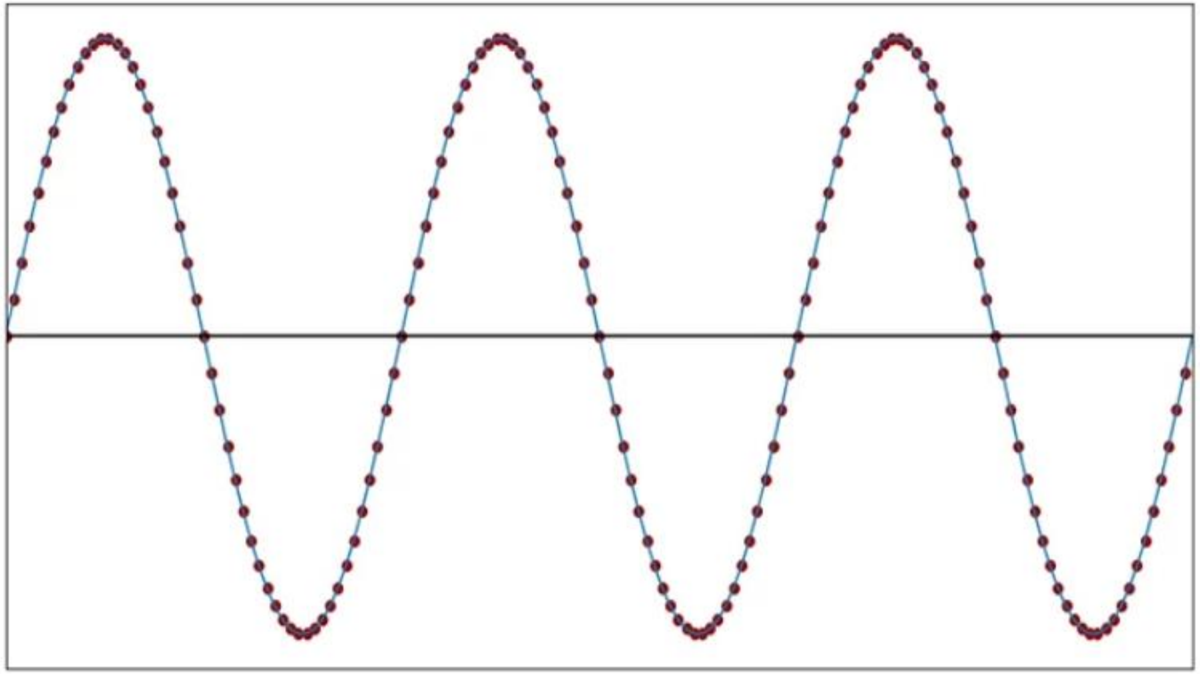


The Nyquist–Shannon Theorem

- If a system uniformly samples an analog signal at a rate that exceeds the signal's highest frequency by at least a factor of two, the original analog signal can be perfectly recovered from the discrete values produced by sampling.

Sampling Theory in the Time Domain

- If we apply the sampling theorem to a sinusoid of frequency f_{SIGNAL}, we must sample the waveform at $f_{SAMPLE} \geq 2f_{SIGNAL}$ if we want to enable perfect reconstruction. Another way to say this is that we need at least two samples per sinusoid cycle. Let's first try to understand this requirement by thinking in the time domain.
- In the following plot, the sinusoid is sampled at a frequency that is much higher than the signal frequency.



The Sampling Theorem

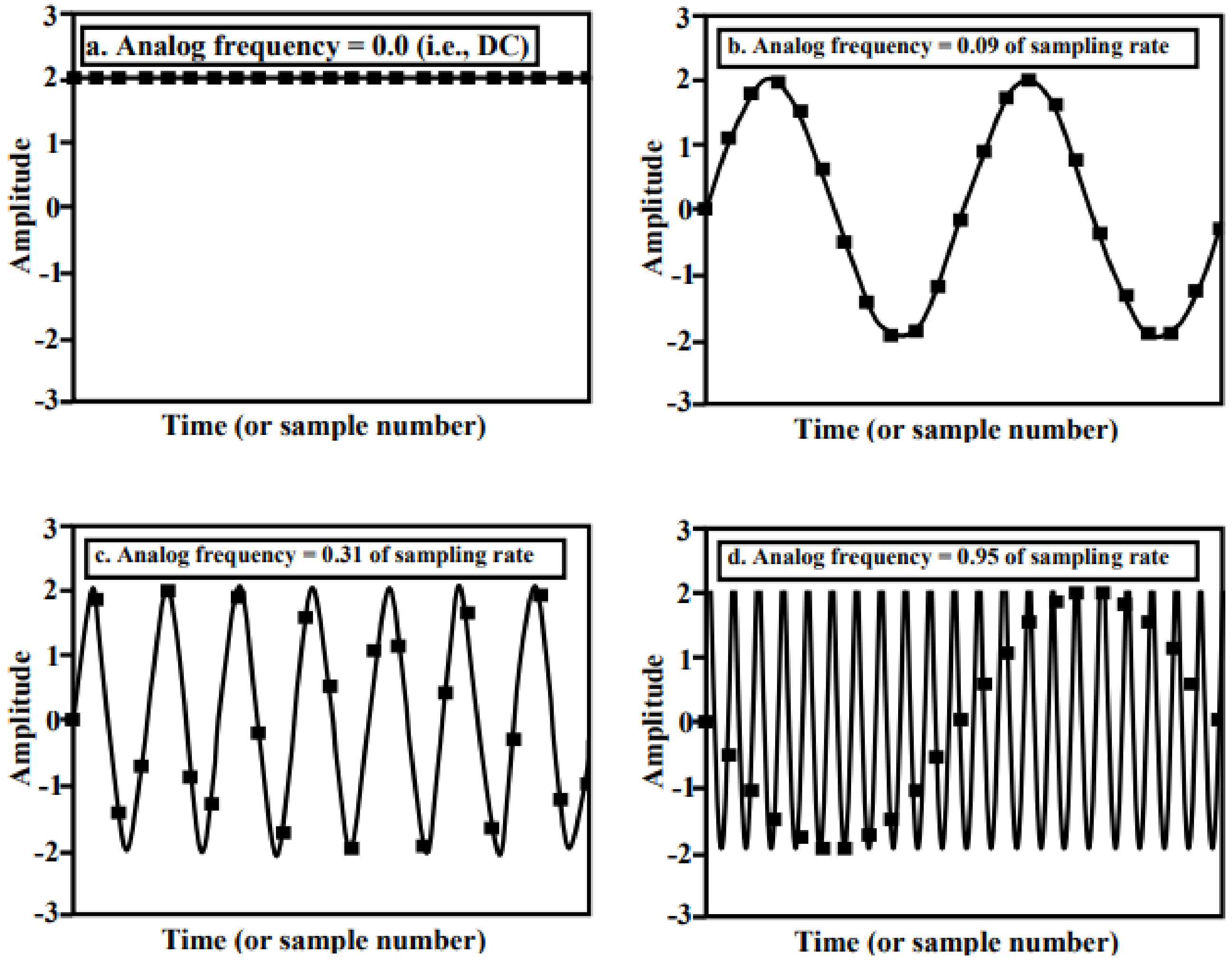
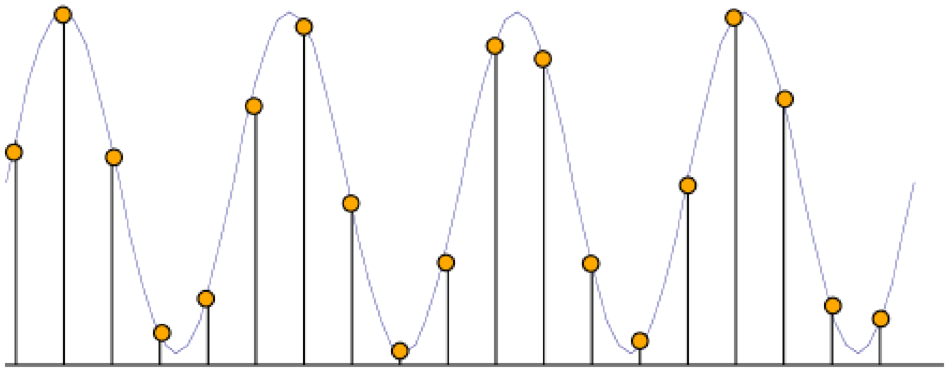
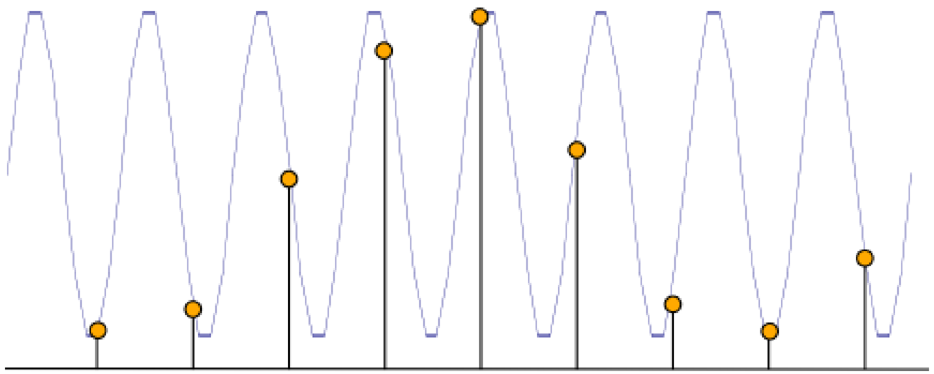


Fig. 18.9 Sampling a sine wave at different frequencies

Aliasing and Antialiasing

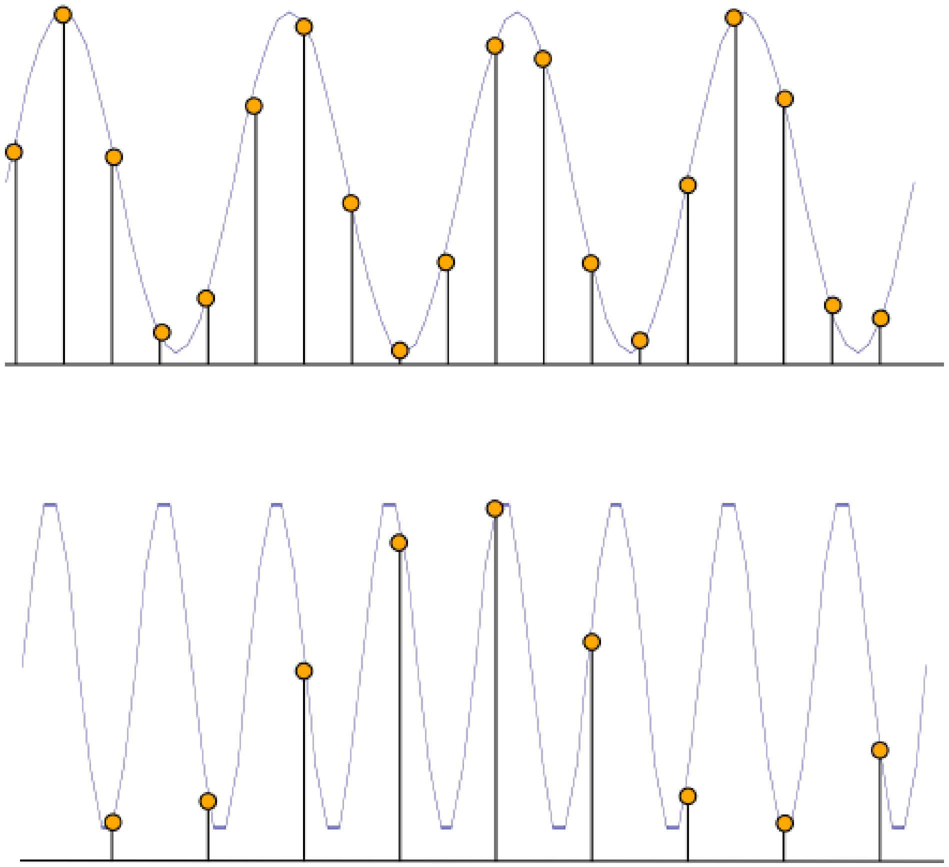
- Suppose you have a signal you want to sample at regular intervals: sampled points are marked in orange
- In the top sinus wave, the sample is fast enough that the reconstructed signals will have the same frequency than the original signal
- In the second wave, instead, the reconstructed wave will be appearing to have a much lower frequency than the original
- This is called an aliased signal





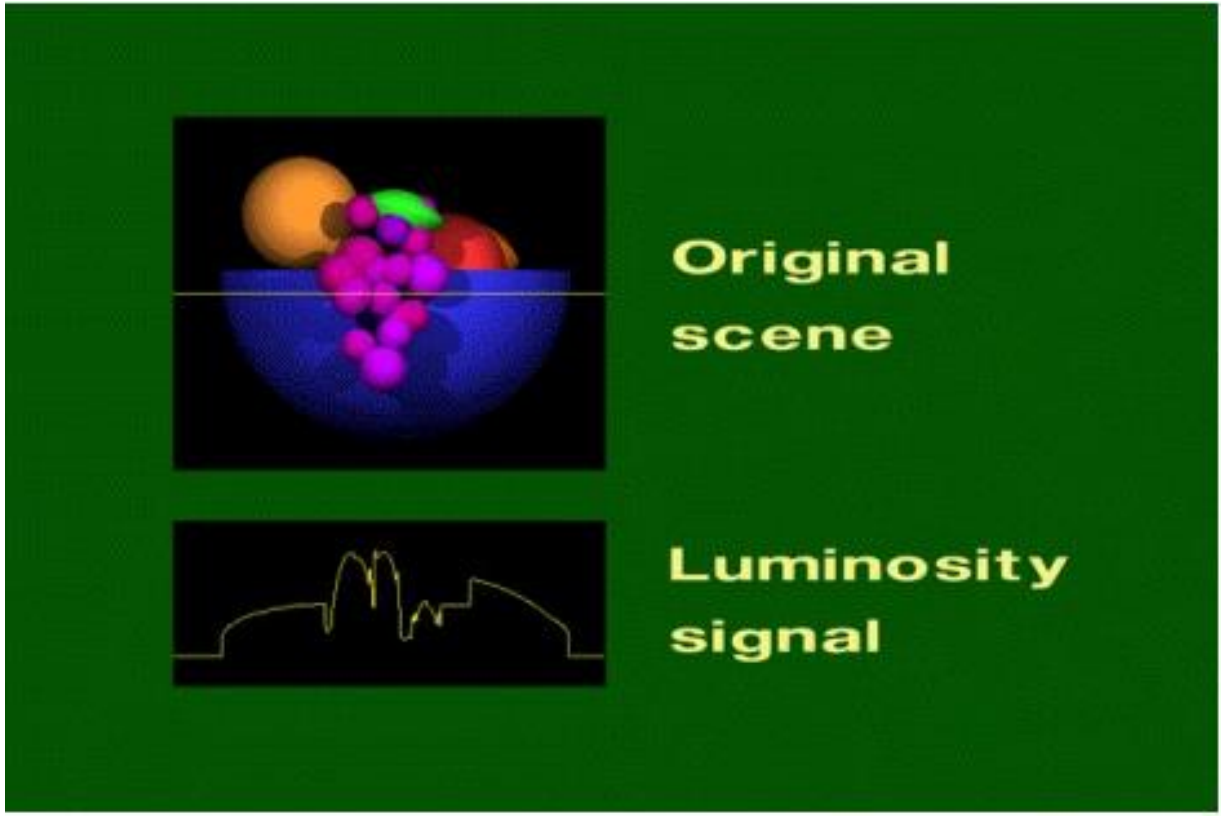
The Nyquist theorem

- In point sampling theory, there is the Nyquist theorem that states that to reconstruct accurately a signal, the sampling rate must be >= 2 times the highest frequency in a signal
- The consequence of this is the fact that music is sampled at 44khz to reproduce the audible spectrum up to 22khz
- Any frequencies over 22khz are removed from the system so as not to have low frequencies due to aliasing

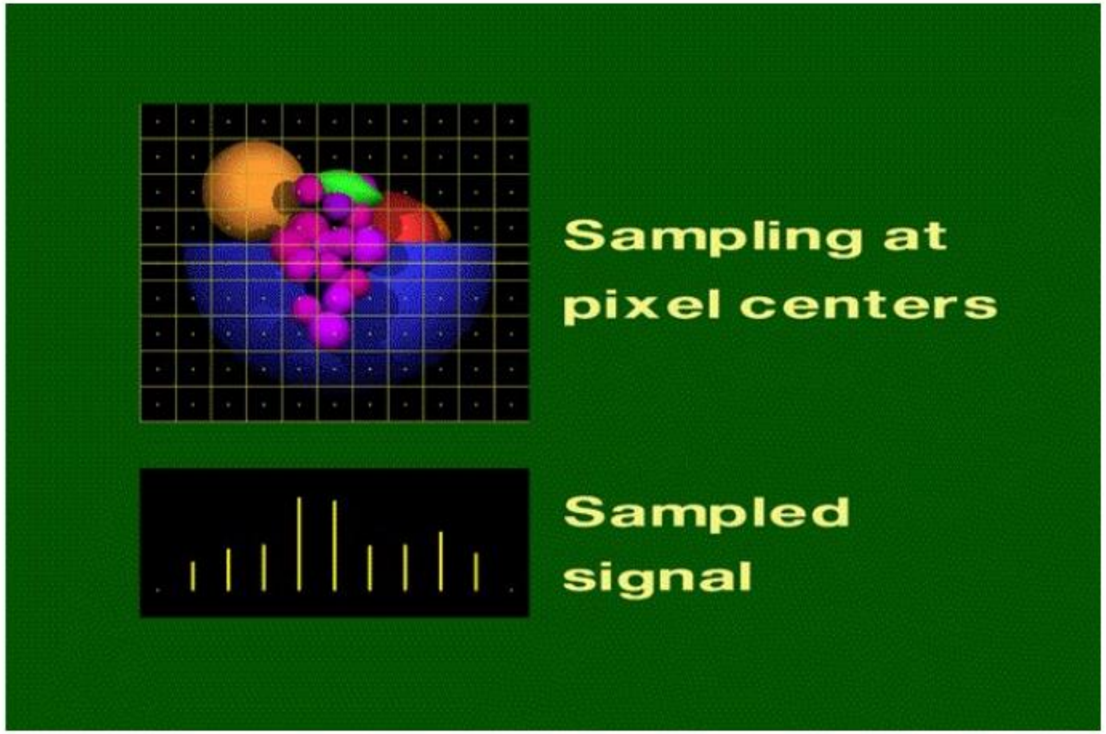


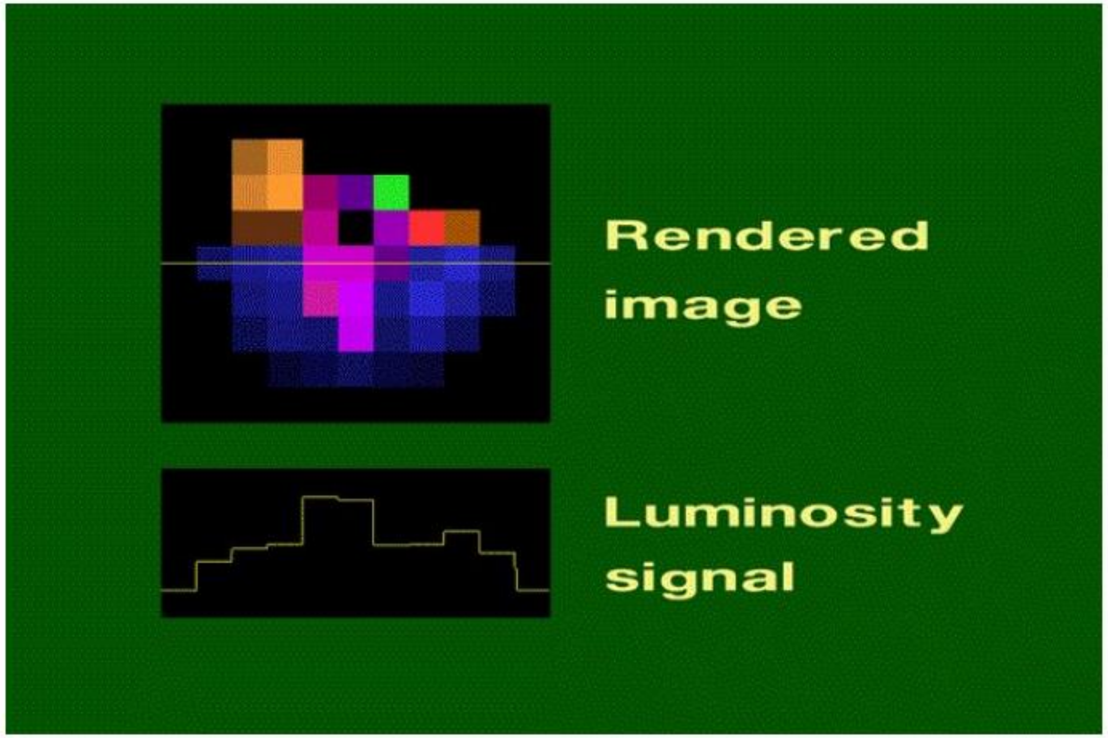
Images as functions

- What does this have to do with graphics?
- An Image can always be seen as a luminosity function F(x,y) of values defined at the pixel centers
- As such it can be seen as the point sampling of a continuous function
- A row of pixels can be therefore seen as a function of the variable x
- Writing pixel values is exactly like sampling the function at the pixel centers



Sampling and Rendering Images



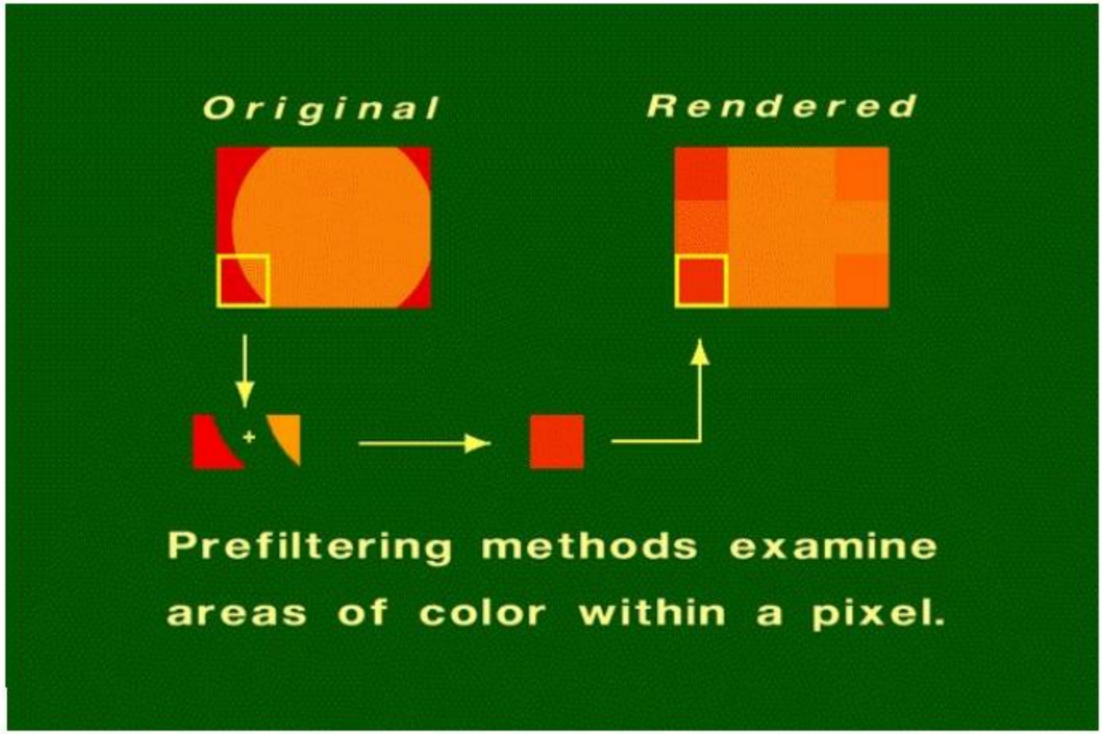


Antialiasing

- Aim of anti-aliasing is to try to avoid the effects of aliasing as much as possible
- There are two main categories of algorithms for doing anti-aliasing
  - prefiltering: treats pixels as an area, and computes pixel color based on the overlap of the scene's objects with a pixel's area.
  - Postfiltering: render the scene at a higher resolution, and compute the pixel value by a (weighted) average of the subpixels (supersampling)

Pre-filtering

- Pixel color is determined by how much percentage of subarea is which color.



Post-filtering

- Pixel color is determined by subsamples:
  - For each pixel, several samples are taken: usually N=4, 9, 16, or 25 subsamples
  - Resulting pixel “sub colors” Ii (i=1,…, N) of the subsamples are then averaged to lead to a pixel color value I

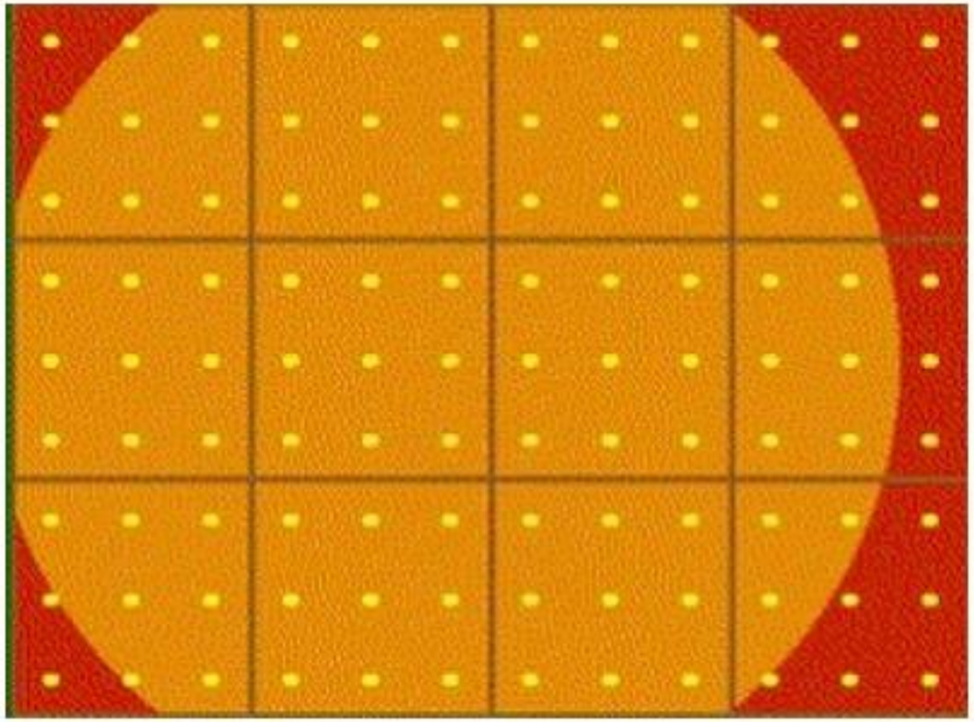
$$I=\Sigma_{i=1,\dots,N} I_i/N$$

  - Sometimes weights w_i are used

$$I=\Sigma_{i=1,\dots,N} w_i I_i/N$$

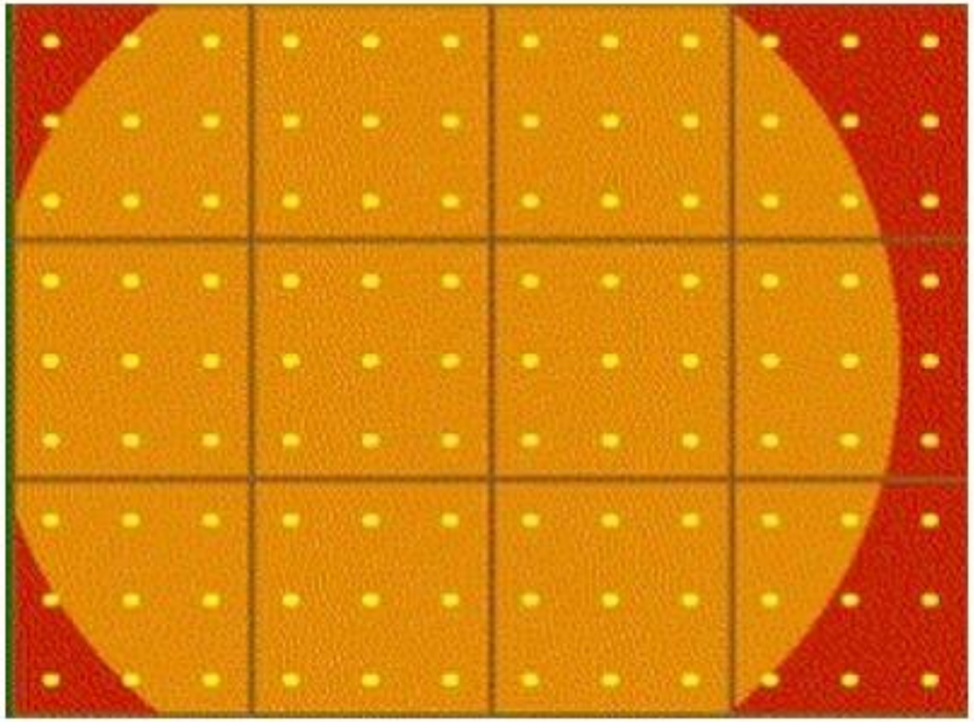
Post-filtering

- There are different ways to determine where to take the subsamples too:
  - Uniform sampling: the samples are taken on a grid (here 9 subsamples)

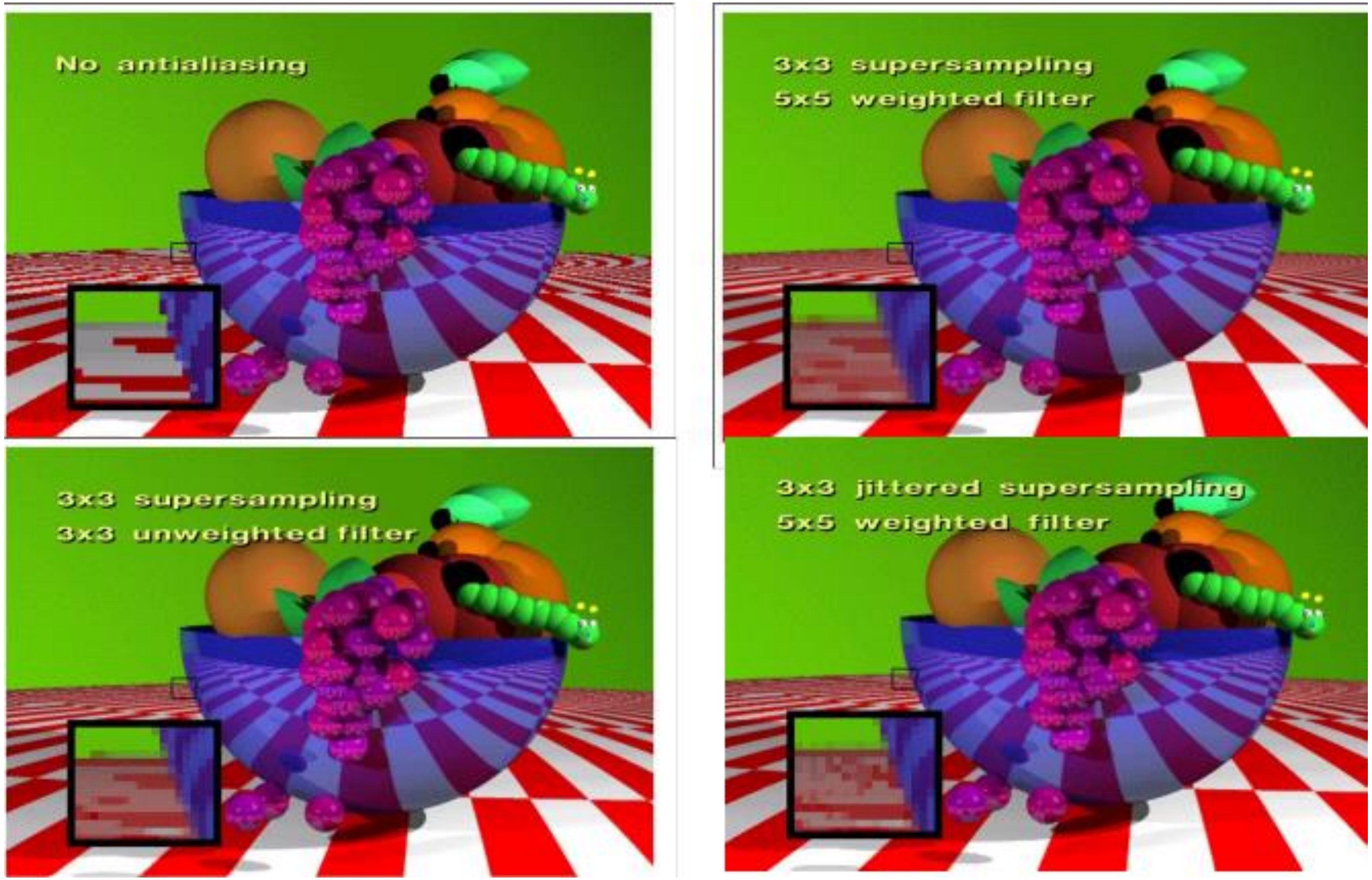


Post-filtering

- There are different ways to determine where to take the subsamples too:
  - Jittered sampling: the samples are centered on a grid, but random values are added to avoid aliasing



Comparison


No antialiasing
3x3 supersampling
5x5 weighted filter
3x3 supersampling
3x3 unweighted filter
3x3 jittered supersampling
5x5 weighted filter

5. Coding and compression of data

5.1 Why do we need coding and compression

Why do we still need coding and compression?

- Compression Technology is employed to efficiently use storage space, to save on transmission capacity and transmission time, respectively. Despite the overwhelming advances in the areas of storage media and transmission networks, it is actually quite a surprise that still compression technology is required.
- One important reason is that the resolution and amount of digital data have increased (e.g. HD-TV resolution) and there are still application areas where resources are limited, e.g. wireless networks. Apart from the aim of simply reducing the amount of data, standards like MPEG-4, MPEG-7, and MPEG-21 offer additional functionalities.

Why do we still need coding and compression?

Three important trends have contributed to the fact that compression technology is as important as it has ever been before – this development has already changed the way we work with multimedia data like text, speech, audio, images, and video which will lead to new products and applications:

- The availability of highly effective methods for compressing various types of data.
- The availability of fast and cheap hardware components to conduct compression on single-chip systems, microprocessors, DSPs, and VLSI systems.
- Convergence of computer, communication, consumer electronics, publishing, and entertainment industries.

What is compression and why do we need it

- Compression is when you condense a file into a smaller space with hopes to make the file smaller (therefore easier to store) while keeping as much quality as possible. The most commonly used types of compression are lossy and lossless.
- We need compression because it saves time if the compressed file is sent as an attachment. A there is a smaller size to send. Also it will save storage space on your computer if you have smaller file sizes.

5.2 Lossless coding and compression

What is Lossless compression

- Lossless compression is when a file is compressed by breaking it up into smaller sections which are then stored. Then the file can be then restored once it is being used or is required to be used. Therefore allowing it to be fully restored.
- One advantage of lossless compression is that the file is fully restored to its original state meaning it does not have a low quality
- One disadvantage of lossless compression is that the file compression takes a lot longer than lossy to compress a file.

Lossless Compression

- In lossless data compression, the integrity of the data is preserved. The original data and the data after compression and decompression are exactly the same because, in these methods, the compression and decompression algorithms are exact inverses of each other: no part of the data is lost in the process. Redundant data is removed in compression and added during decompression. Lossless compression methods are normally used when we cannot afford to lose any data.

5.2.1 arithmetic coding

Arithmetic coding

- In Huffman coding we have a correspondence between a single symbol and its codeword – which is the main reason for its suboptimality. Arithmetic coding uses a single codeword for an entire sequence of source symbols of length m. In this manner the restriction to integer-valued bits per symbol values can be avoided in an elegant way. A drawback however is that similar sequences of source symbols can lead to entirely different codewords

Arithmetic coding

- Arithmetic coding is a data compression technique that encodes data (the data string) by creating a code string that represents a fractional value on the number line between 0 and 1.
- The coding algorithm is symbolized recursively; i.e., it operates upon and encodes (decodes) one data symbol per iteration or recursion. On each recursion, the algorithm successively partitions an interval of the number line between 0 and I, and retains one of the partitions as the new interval. Thus, the algorithm successively deals with smaller intervals, and the code string, viewed as a magnitude, lies in each of the nested intervals.
- The data string is recovered by using magnitude comparisons on the code string to recreate how the encoder must have successively partitioned and retained each nested subinterval. Arithmetic coding differs considerably from the more familiar compression coding techniques, such as prefix (Huffman) codes.
- Also, it should not be confused with error control coding, whose object is to detect and correct errors in computer operations. This paper presents the key notions of arithmetic compression coding by means of simple examples

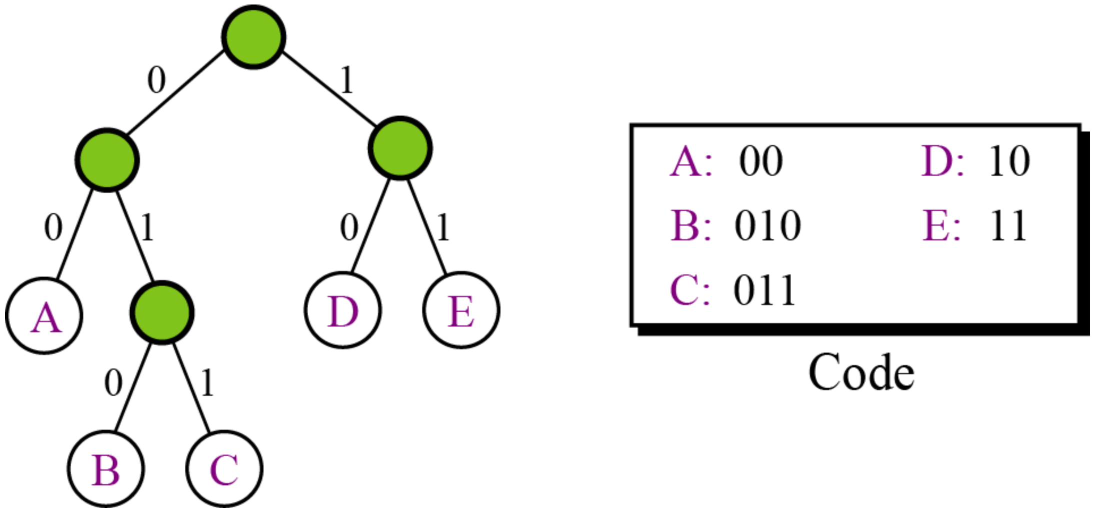
5.2.2 Huffman coding

Huffman coding

- Huffman coding assigns shorter code to symbols that occur more frequently and longer codes to those that occur less frequently. For example, imagine we have a text file that uses only five characters (A, B, C, D, E). Before we can assign bit patterns to each character, we assign each character a weight based on its frequency of use. In this example, assume that the frequency of the characters is as shown in teh table below.

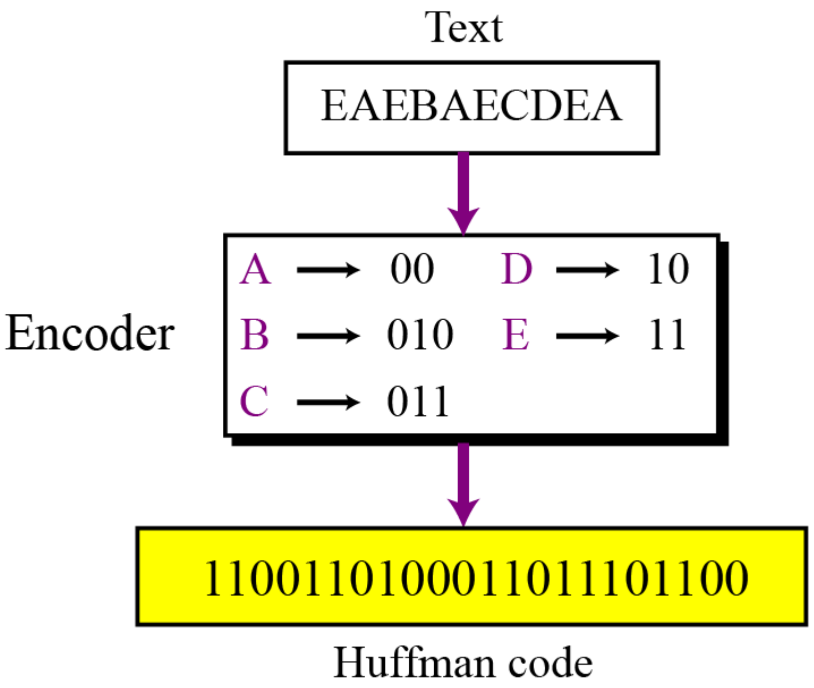
Character	A	B	C	D	E
Frequency	17	12	12	27	32

A character’s code is found by starting at the root and following the branches that lead to that character. The code itself is the bit value of each branch on the path, taken in sequence.



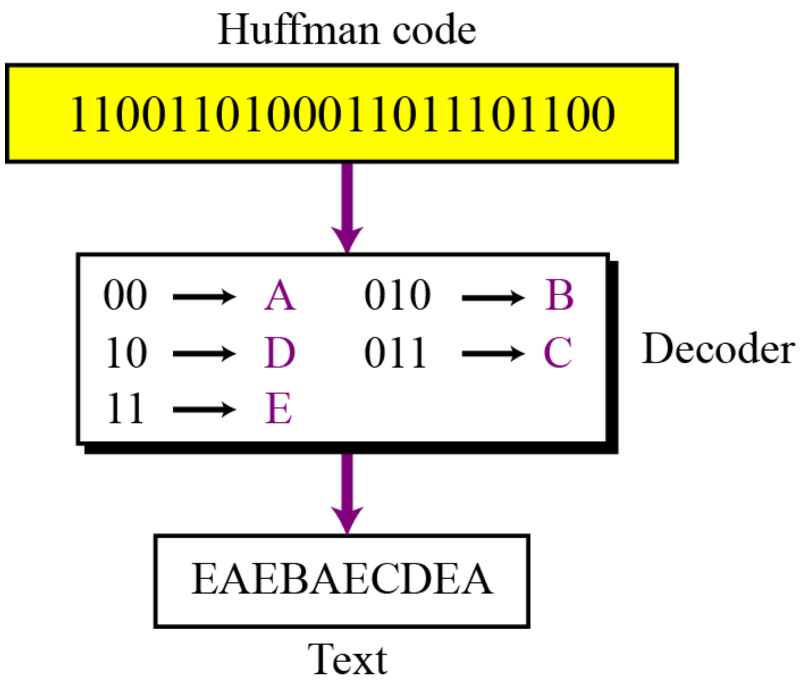
Encoding

- Let us see how to encode text using the code for our five characters. Figure below shows the original and the encoded text.



Decoding

- The recipient has a very easy job in decoding the data it receives. The figure below shows how decoding takes place



Huffman Encoding

- Static Huffman coding assigns variable length codes to symbols based on their frequency of occurrences in the given message. Low-frequency symbols are encoded using many bits, and high-frequency symbols are encoded using fewer bits.
- The message to be transmitted is first analyzed to find the relative frequencies of its constituent characters.
- The coding process generates a binary tree, the Huffman code tree, with branches labeled with bits (0 and 1).
- The Huffman tree (or the character codeword pairs) must be sent with compressed information to enable the receiver to decode the message.

5.2.3 Runlength coding

Run-Length-Encoding

- Run-length encoding (RLE) is a very simple form of data compression in which a stream of data is given as the input (i.e. "AAABBCCCC") and the output is a sequence of counts of consecutive data values in a row (i.e. "3A2B4C"). This type of data compression is lossless, meaning that when decompressed, all of the original data will be recovered when decoded. Its simplicity in both the encoding (compression) and decoding (decompression) is one of the most attractive features of the algorithm
- There are many different compression methods. Compression methods can be categorized as being lossless or lossy. With a lossless compression method, the original data can be restored exactly as it was before it was decompressed, whereas when a lossy compression method is used some of the original data is lost during the compression process and cannot be restored when the data is decompressed.
- There are times when lossy compression is the best choice – either when significantly smaller file size is desired or if the data loss is not noticeable, eg there are many sounds that people cannot hear and if compression results in the loss of some of these sounds then the data loss is not important.

Bit Level RLE

- Bit level RLE is effective when one bit is used to represent the colour of each pixel, ie in a monochrome image. Here a single byte represents the value of a pixel and the run length. Within the 8 bits, the left-most bit identifies the colour (eg 0 = white and 1 = black) the next 7 bits identify the run length (runs that are longer than 127 need to be broken down into a number of 127 long runs plus a run to represent the pixels 'left over').

Byte Level RLE

- Each pixel (colour) is represented by a single byte giving 256 possibilities. The colours themselves would be held within a colour palette table which would be stored in the image file. The colour palette is arbitrary and determined by factors such as hardware, image composition or file type. The image data is then represented by pairs of bytes representing the run length and then the index of the run colour in the color palette table

Run-length encoding

- Run-length encoding is probably the simplest method of compression. It can be used to compress data made of any combination of symbols. It does not need to know the frequency of occurrence of symbols and can be very efficient if data is represented as 0s and 1s.
- The general idea behind this method is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences.
- The method can be even more efficient if the data uses only two symbols (for example 0 and 1) in its bit pattern and one symbol is more frequent than the other.

Pixel Level RLE

- Each pixel is represented by three bytes, for an RGB bitmap. Each pair would consist of a run-length byte, followed by the three bytes that represent the pixel color.
- The metadata for a bitmap will include the number of rows and number of columns (in this case 8x8), so it is safe for the RLE to 'run over a row, ie reading from the top-left pixel.
- Because of this, RLE is only good for certain types of data and applications. For example, the Pixy camera, which is a robotics camera that helps you easily track objects, uses RLE to compress labeled video data before transferring it from the embedded camera device to an external application. Each pixel is given a label of "no object", "object 1", "object 2", etc. This is the perfect encoding for this application because of its simplicity, speed, and ability to compress the low-entropy label data.

5.2.4 Dictionary compression

Dictionary compression

- Basic idea: the redundancy of repeated phrases in a text is exploited (recall: in Runlenth encoding, repeated identical symbols can be efficiently coded). Here we encode repeatedly occuring symbol strings in arbitrary files in an efficient manner. A frontend selects strings and information for compression and decompression is stored in a dictionary. The encoder and the decoder need to have access to a copy of this dictionary of course. In this manner, entire strings are replaced by tokens (i.e. codewords). The basic algorithmic elements are lookup tables, and the corresponding operations are very fast.

5.3 Lossy coding and compression

What is Lossy compression

- Lossy compression is where a file is compressed by taking out unimportant information from a file. For example, any unnecessary information within the file will be removed to make the folder smaller.
- One advantage of lossy compression is that it is relatively easy to compress files that have a lot of useless information that is not required. Which makes it faster than lossless compression
- However one disadvantage of lossy compression is that once you have compressed the file it is impossible to restore the file to its original state. Due to information being taken away from a file and it can not be bought back.

5.3.1 Quantization

Quantization

- When changing the description of a signal from using a large alphabet to using a smaller alphabet of symbols we apply quantization. Obviously, when reversing the process, perfect reconstruction is no longer possible since two or more symbols of the larger alphabet are mapped to a single symbol of the small alphabet. Often, quantization errors are the only actual errors caused by lossy compression (apart from rounding errors in transforms).

Quantization

- Example: suppose we have stored a grayscale image with 8bit/pixel (bpp) precision, which means that for each pixel, we may choose among 256 different grayscales. If we apply a quantization to 4 bpp, only 16 grayscale can be used for a pixel. Of course, it is no longer possible to re-transform the 4 bpp into the original bit-depth without errors. The information is lost.

5.3.2 Transform coding

Transform coding

- Most standardized lossy media compression schemes are based on transform coding techniques. Typically, transform based coding consists of three stages.

1. Transformation
2. Quantisation: a variety of techniques (as discussed before) can be applied to map the transform coefficients to a small alalphabetthe size of the latter being selected corresponding to the target bitrate. Most rate-control procedures are applied in this stage.
3. Lossless encoding of coefficients: a variety of techniques (as discussed before) is applied in order to encode the quantized transform coefficients close to the entropy bound. Typically, a combination of run length (zero-runs !) and Huffman coding (for the older standards like JPEG, MPEG-1,2,4) or arithmetic coding (for the more recent standards like JPEG2000 H.264) is used.

Transform coding

- The aim of transformation is to change the data of a signal in a way that it is better suited for subsequent compression. In most cases, so-called "integral-transforms" are used, which are based (in their contineous form) on applying integral operators. When applying transforms, the concept of projecting the data onto orthogonal basis functions is used.

5.3. Compression Lossy and Lossless

Similarities between the two compresion

- They both compress files into a smaller file
- They are both program based
- Tehy both can be programmed

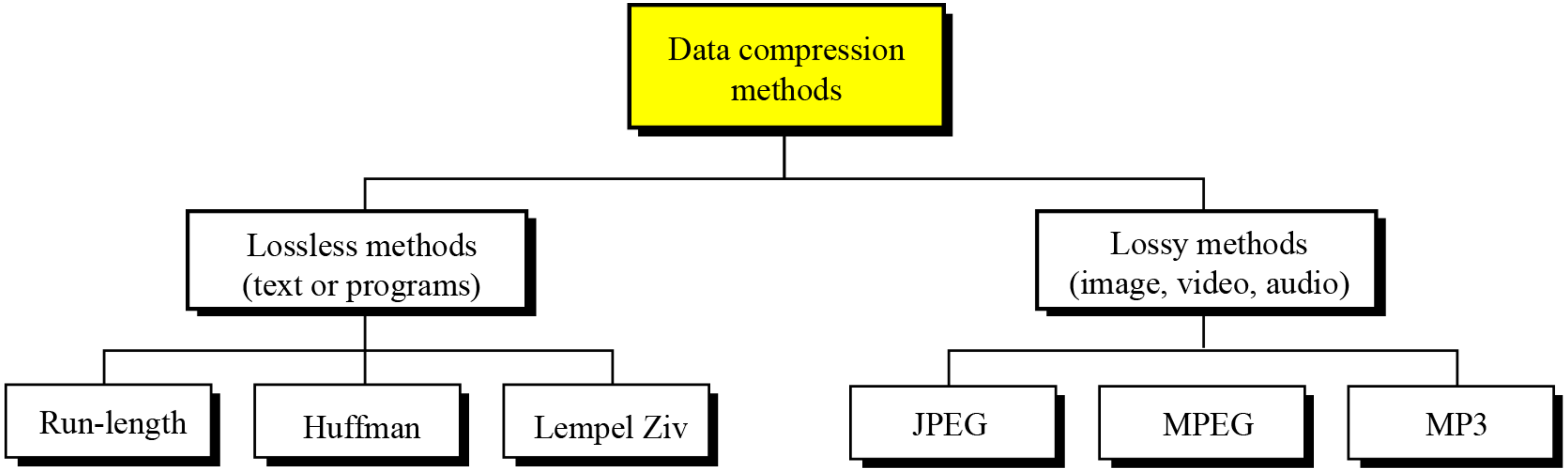
Differences between the two compressions

- Lossy:
  - Unimportnat data is removed
  - Faster than lossless
  - Lossy:
- Lossless:
- Data is kept
- Slower than lossy

Examples of using lossy and lossless files

- Lossy:
  - JPEG
  - MP3
  - MPEG
- Lossless
  - Hufman
  - LZW
  - PNG
  - FLAC (Free Lossless Audio Codec)

- Data compression implies sending or storing a smaller number of bits. Although many methods are used for this purpose, in general these methods can be divided into two broad categories: lossless and lossy methods.



Usage of coding algorithms

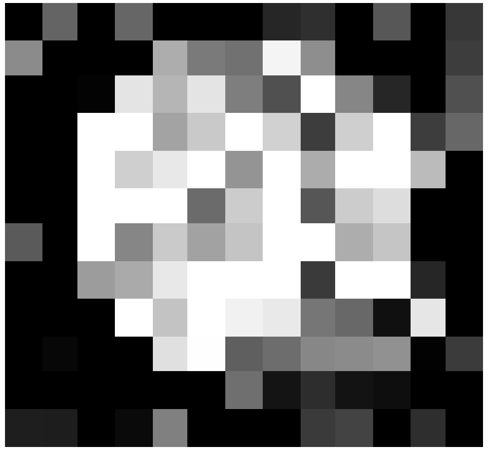
Rle:	**bitmap, game industry**
Lzw:	zip, rar
Hufman:	often combined with lossy compression algorithms

6. Understanding Digital Images

Digital imaging

- Digital images: Crafting Analog Precision
  - The art of digitally storing images involves creating a representation remarkably close to the original analog.
  - Precision in measurements (e.g., a thermometer displaying 37.4319 degrees) when repetitively recorded captures the essence of the analog original.
- Pixel Arrangement and Realism:
  - A digital image comprises an arrangement of points known as pixels.
  - Each pixel carries a grayscale or color value (refer to Figure: B.4).
  - Analogous to coloring a grid, where each grid square can be filled with only one color.
  - With small enough squares and a diverse range of colors, the resulting image faithfully mirrors reality.

0	100	0	102	0	0	0	38	47	0	87	0	55
139	0	0	0	173	122	113	244	142	0	0	0	61
0	0	3	228	181	228	126	80	255	134	38	0	80
0	0	255	255	163	201	255	209	61	207	255	61	103
0	0	255	207	232	255	148	255	172	255	255	187	0
0	0	255	255	255	107	204	255	86	204	221	0	0
90	0	255	134	202	162	196	255	255	173	197	0	0
0	0	156	170	232	255	255	255	58	255	255	38	0
0	0	0	255	195	255	241	233	118	104	16	230	0
0	7	0	0	224	254	95	109	135	139	145	2	59
0	0	0	0	0	0	111	20	45	19	14	0	0
30	29	0	10	128	0	0	0	58	67	0	46	0



Visualizing a digital image as an array of points called pixels. Illustrated using a grayscale image example.

Image **Acquisition**

- Image Acquisition: Capturing Images
  - Image acquisition can be categorized into analog and digital domains.
  - The transition between analog and digital occurs through Analog-to-Digital (A/D) conversion.
- Analog Image Acquisition:
  - Light stimuli are detected using a sensor in devices like CCD cameras, digital cameras, and mobile devices.
  - The resulting analog signal from CCD cameras requires external A/D conversion, while digital cameras and mobile phones integrate this process.
- Importance of Digitalization:
  - Digitalization is crucial for subsequent processing using computers.
  - Enables a range of processing and manipulation possibilities.
  - The outcomes can be saved on various storage media, accommodating different formats and sizes

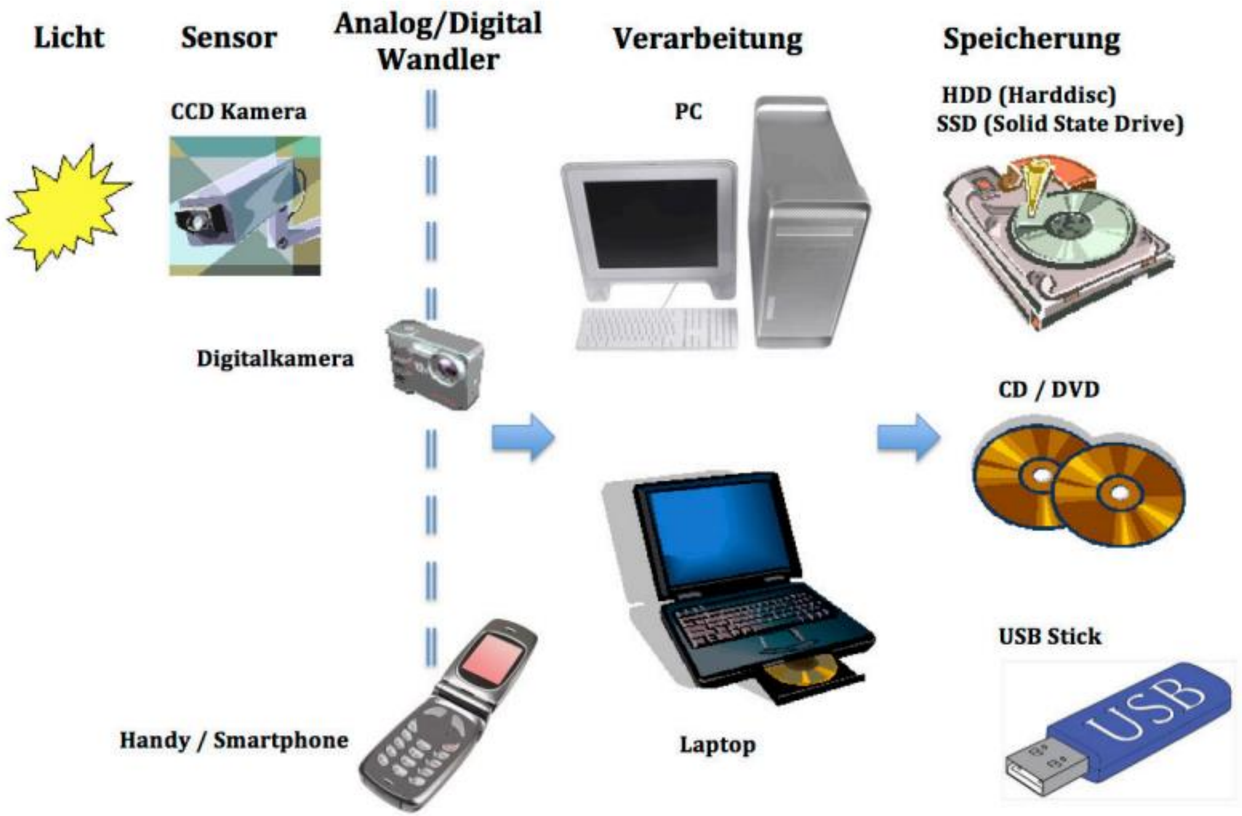


Abbildung B.6: Bildaufnahme

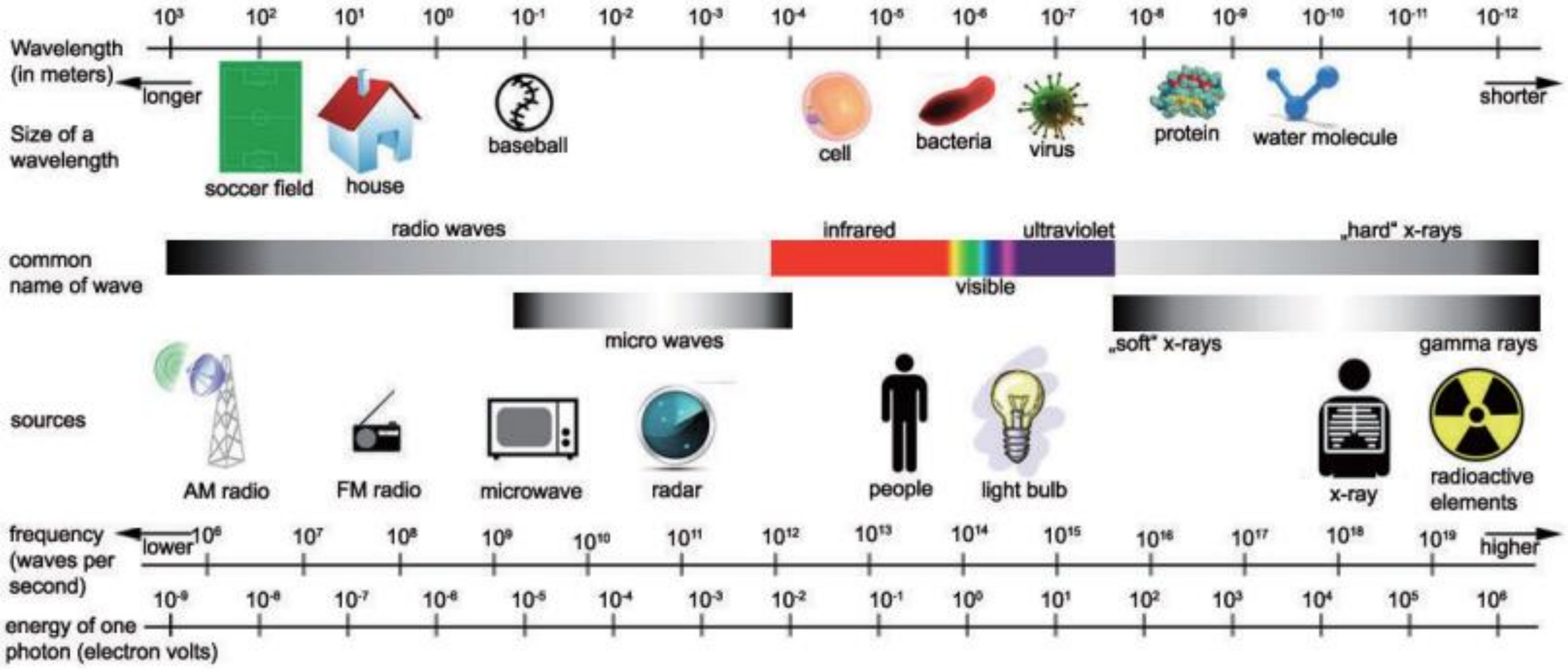
Image acquisition

The Nature Of Light

- Dual Nature of Light: Particles and Waves
  - Light exhibits both particle and wave characteristics, explored through experimental physics.
  - This phenomenon is known as "Wave-Particle Duality" in quantum physics.
- Photon Particle Model:
  - Light particles are called photons.
  - Photons are massless and travel at the speed of light (c = 299,792.46 km/s) regardless of the medium or observer's motion.
- Diverse Wavelengths of Light:
  - Light encountered in technology and nature possesses various wavelengths.
  - These wavelengths span both the visible and invisible spectra.
  - Human eyes perceive wavelengths from approximately 380 nm (red) to 790 nm (blue).
  - Greatest sensitivity in the green range (≈ 480 nm - 560 nm).

Light Wavelength Ranges

- Light Wavelengths < 380 nm:
  - Include regions like:
    - Ultraviolet Radiation (1 nm - 380 nm)
    - X-ray Radiation (1 pm - 10 nm)
    - Gamma Radiation (< 5 pm)
- Light Wavelengths > 770 nm:
  - Encompass regions like:
    - Infrared Radiation (780 nm - 1 mm)
    - Radar Signals (1 m - 10 m)
    - Radio Waves (1 m - 10 km)



Light Spectrum

Sensor Technology

- Quantum Fluctuations and Noise:
  - Quantum fluctuations can be viewed as noise.
  - Signals with intensity and wavelength variations may be masked by noise signals, deteriorating the quality of the useful signal.
- Signal-to-Noise Ratio (SNR):
  - Signal quality measured by Signal-to-Noise Ratio (SNR).
  - SNR defines the ratio of the average power of the useful signal to the average power of the interfering noise signal.
- Balancing Resolution and Averaging:
  - Signal quality can improve through spatial averaging, but at the cost of spatial resolution.
  - Temporal averaging enhances signal quality at the expense of temporal resolution.
  - Striving for a compromise between these aspects is the objective.

Erhöhung der Zuverlässigkeit (SNR) einer Intensitäts/Wellenlängenschätzung:

Mittelung über grössere Flächen	Mittelung über grössere Zeitintervalle
Verlust räumlicher Auflösung	Verlust zeitlicher Auflösung

Kompromiss zwischen { der Genauigkeit der Intensitäts/Wellenlängen Messung und der räumlich/zeitlichen Auflösung!

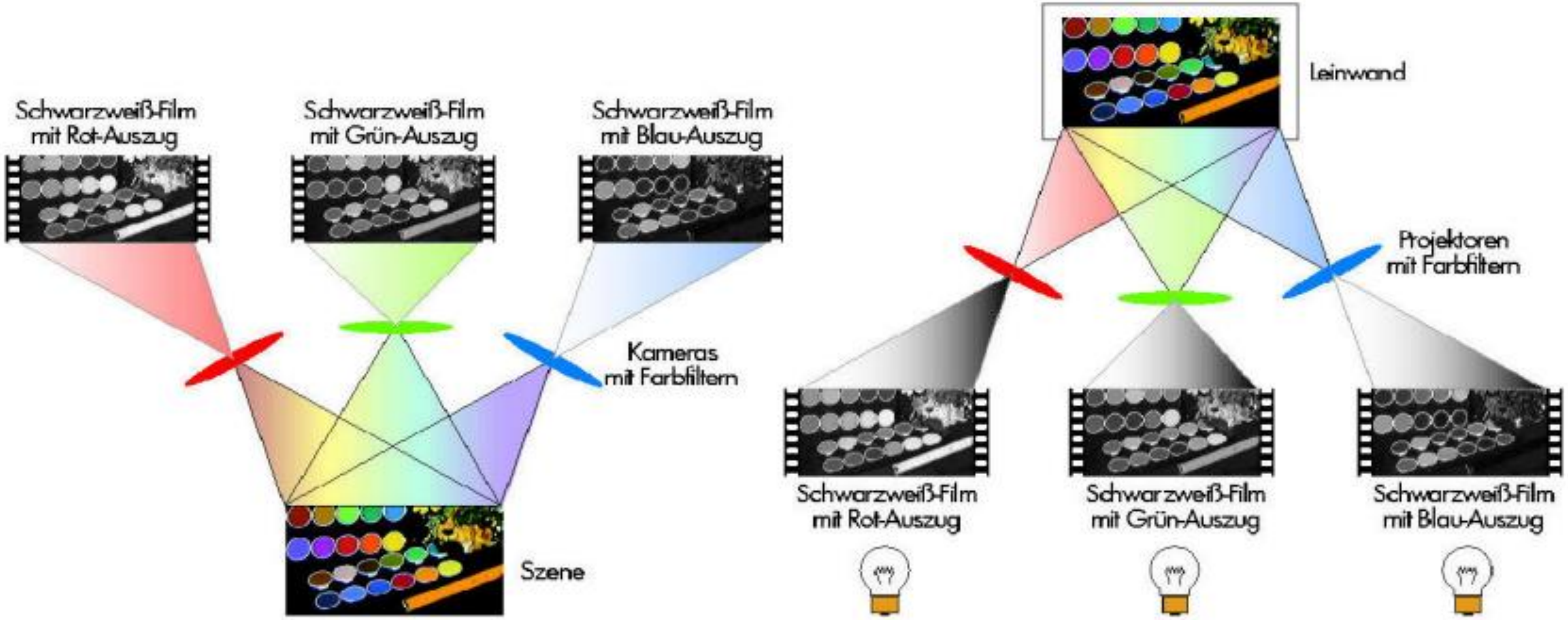
Improving the quality of useful signals

Color Vision and Color Cameras

- Human Color Perception:
  - Human eyes have brightness receptors (rods) and three types of color receptor cells (cones) on the retina.
  - Cones are sensitive to different wavelengths (colors), broadly categorized into red, green, and blue ranges.
- Color Perception and Mixing:
  - Eyes perceive color as a combination of primary colors.
  - Additive mixing involves red, green, and blue (RGB) as primary light colors.
  - Subtractive mixing uses cyan, yellow, and magenta as primary colors.
- RGB Format and Color Representation:
  - TVs use electron beams triggering phosphors to emit red, green, and blue light, creating images.
  - RGB format encompasses colors producible by additive mixing of red, green, and blue.
  - Common format for storing images in computers.

Color Cameras and Imaging

- Figure B.9 (Left): Color Filtering in Cameras:
  - Scene captured by camera's color filters dividing it into red, green, and blue components.
  - Presenting these as black and white images.
- Figure B.9 (Right): Color Projection:
  - Projecting these images using corresponding color filters creates additive mixing of individual color components, replicating pure spectral colors.



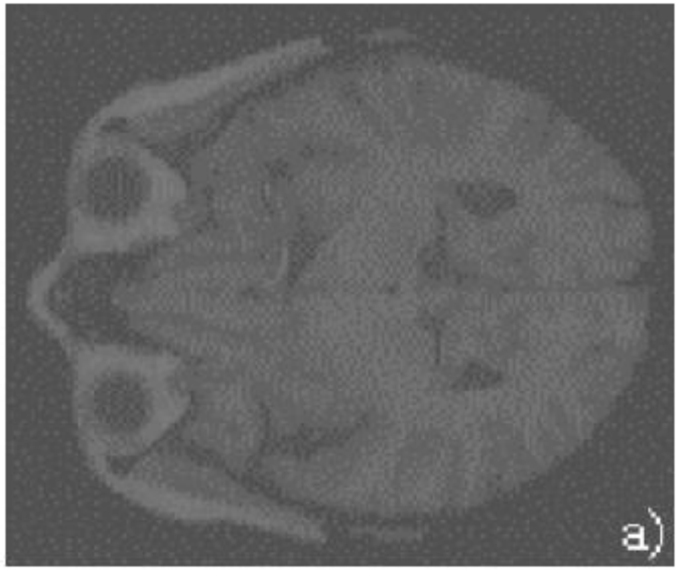
Additively mixed light from variable red, green, and blue components causes im human eye the same color impression as a 'pure' spectral color.

Image quality: Defining Parameters

- Image quality:
  1. Sharpness
  2. Contrast
  3. Resolution
  4. Disturbances/Noise
- Sharpness (B.10.1):
  - Sharpness refers to the accuracy of representing image details.
  - Measured by edge profile; steeper slope = sharper edges, flatter slope = blurrier image.
- Contrast (B.10.2):
  - Contrast is the difference between maximum and minimum pixel values.
  - Reflects local brightness change; the ratio of object's average brightness to background brightness.
  - High contrast utilizes intensity levels effectively; derived from histograms.
- Resolution (B.10.3):
  - Resolution signifies pixel size in the real world.
  - Units include dpi (dots per inch), lpi (lines per inch), or pixels per kilometer (satellite images).

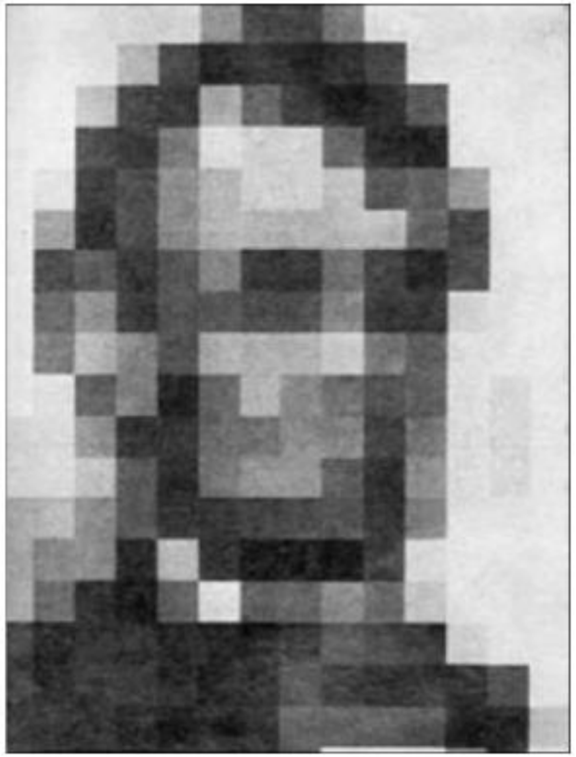
In the left image, the letter “i” is sharpened, for comparison, a Blurred image of the same subject can be seen on the right side of the figure (edges are smudged, Details are not exactly recognizable)

Sectional images of a human brain



(a) Sectional image of a human brain with low contrast (value range: 82 to 255)

(b) Compared to Figure B.11(a), a higher-contrast image of the same subject can be seen (object stands out clearly from the background - value range: 0 to 255)



Leon Harmon of Bell Labs: picture of Lincoln (252 pixels), The Recognition of Faces, Scientific American, (Nov. 1973).

Artifacts and Noise

- Artifacts and Noise
- Abtastung und Objektgrösse:
  - Mismatched resolutions between image and display create artifacts.
  - Illustrated in Figure B.13 (example: image pixel size $1mm^2$, display pixel size $9mm^2$).
- Artifact Mitigation:
  - Look-up Table (LUT) adapts gray values for display.
  - Example: Reduce pixel count by 3, create average values, and adjust with LUT.
- Artifact Effects (B.10.4):
  - Smoothing due to averaging; loss of detail (Figure B.14).
  - Noise: Image quality can degrade from disturbances.
  - Gaussian noise (Figure B.15(a)), additive noise (Figure B.15(b)), multiplicative noise (Figure B.15(c)), pixel disturbances (Figure B.15(d)).

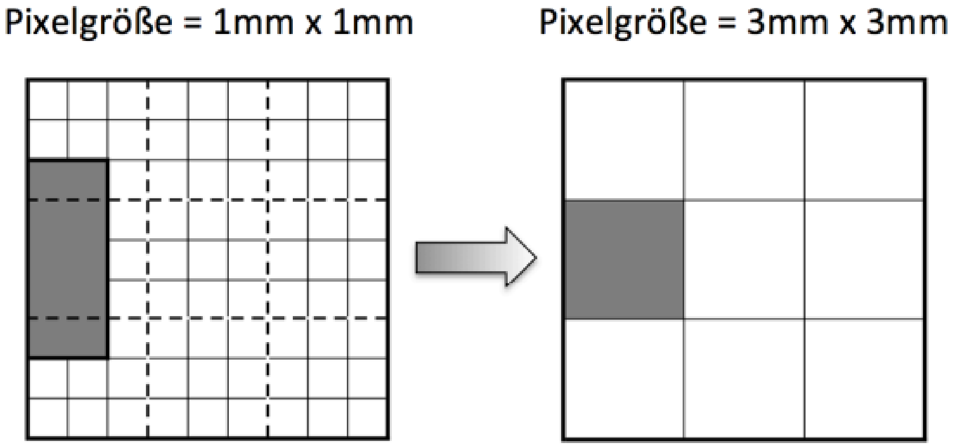
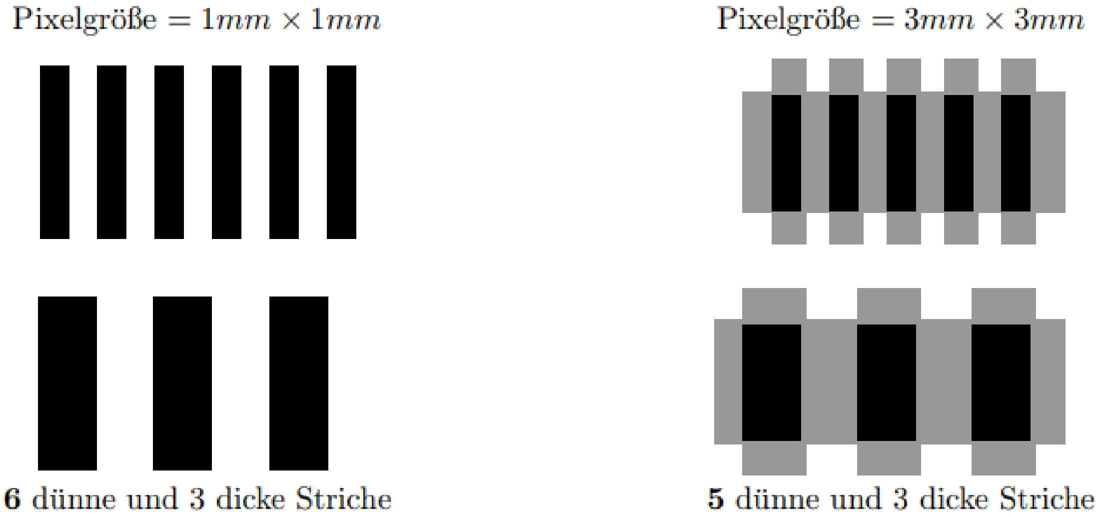


Figure B.13: Problem of adapting different resolutions



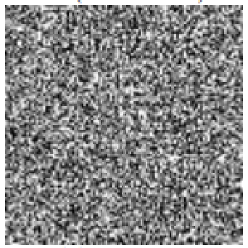
Problem of adapting different resolutions

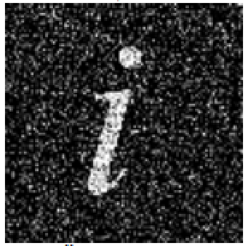
Tabelle B.1: LUT (oben) Veranschaulichung der Kontraststreckung (unten)

LUT (Look up table)		
Farbe	Grauwerte	Gewünschte Ausgabegrauwerte
weiss	3	3
	2	3
	1	0
schwarz	0	0

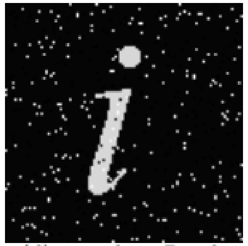
3 Striche:													
Grauwerte:	3	3	3	3	0	3	0	3	0	3	3	3	3
Mittelwerte:	-	3	3	2	2	1	2	1	2	2	3	3	-
Kontrast gestreckt:	-	3	3	3	3	0	3	0	3	3	3	3	-
2 Striche:													

Noise	**additiv**	**multiplikativ**	**Salz (255) und Pfeffer (0)**
.	.	.	0 if $N < \epsilon$
.	.	.	255 if $N > 255 - \epsilon$
N (Gaussian)	$I + N$	$I .* N$ (speckle)	I sonst


(a) Störung


(b) des Übertragungskanal


(c) signalabhängig


(d) einzelner Pixel

Types of image interference or image noise

7. Medical Imaging

8.1MRI

Magnetic Resonance Imaging (MRI)

- MRI Overview:
    - Magnetic Resonance Imaging (MRI) is a powerful medical imaging technique widely used to visualize internal structures of the body.
    - It relies on the interaction between strong magnetic fields and radio waves to create detailed images.
- Principles of MRI:
    - MRI is based on the behavior of hydrogen atoms (protons) within the body.
    - When placed in a magnetic field, hydrogen nuclei align themselves with the field.
    - Radiofrequency pulses are applied, causing these aligned nuclei to emit signals as they return to their equilibrium state.
    - The signals are captured and processed to generate detailed cross-sectional images.
- Clinical Applications:
    - Neuroimaging: Used to visualize the brain and spinal cord, aiding in diagnosing tumors, strokes, and neurological disorders.
    - Musculoskeletal Imaging: Assesses joint and musculoskeletal conditions, such as injuries, arthritis, and torn ligaments.
    - Cardiovascular Imaging: Provides detailed images of the heart and blood vessels, aiding in diagnosing heart conditions.
    - Abdominal and Pelvic Imaging: Detects tumors, infections, and abnormalities in organs like the liver, kidneys, and reproductive organs.
- Contrast Agents:
    - Some MRI exams use contrast agents, such as gadolinium-based agents, to enhance the visibility of certain structures.
    - These agents highlight blood vessels, tumors, and areas of inflammation, providing additional diagnostic information.

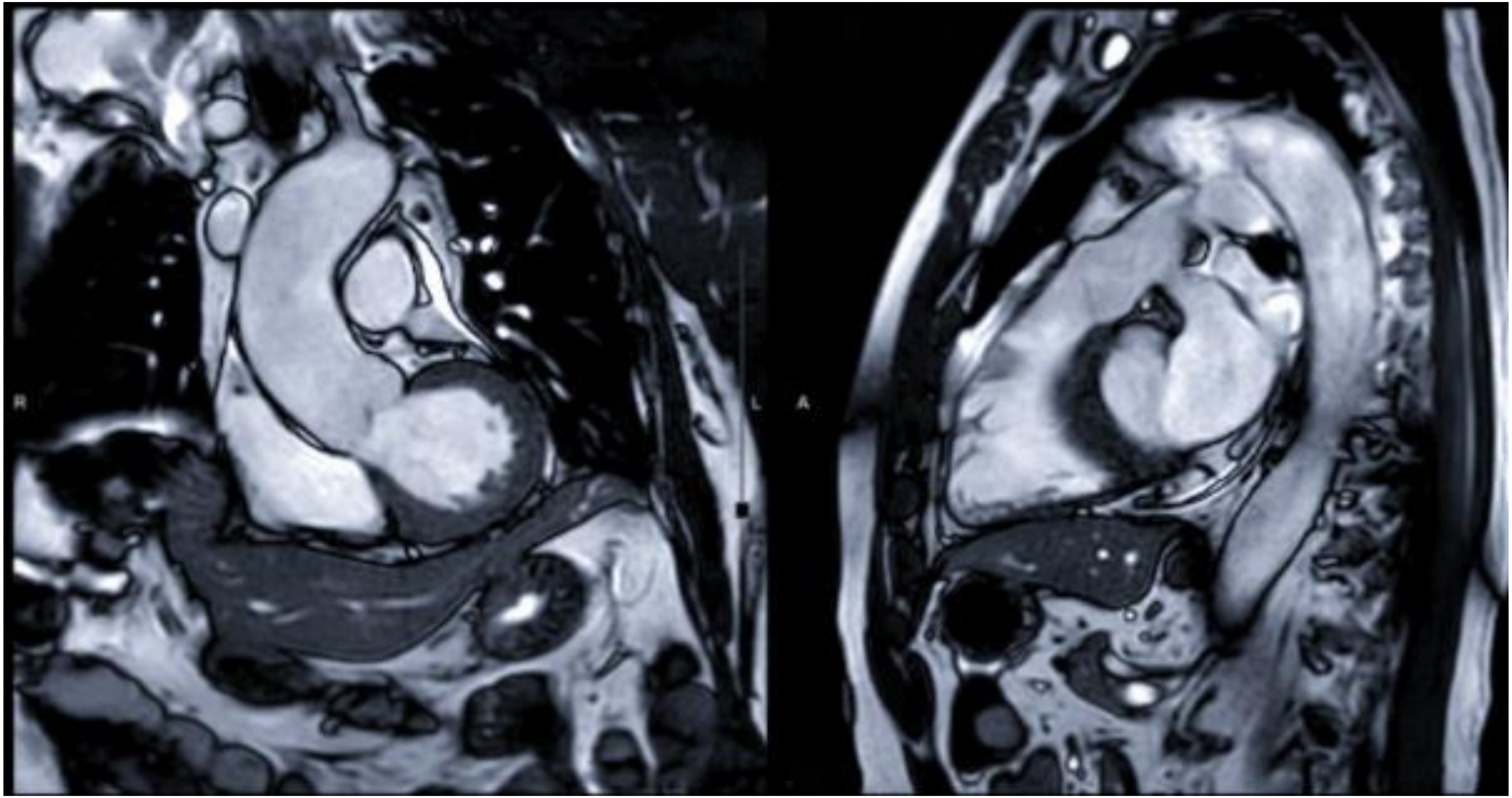
Pros and cons

Advantages

- Non-Ionizing Radiation: Unlike X-rays and CT scans, MRI does not use ionizing radiation, making it safer for repeated imaging.
- Soft Tissue Contrast: MRI provides exceptional contrast between different soft tissues, making it suitable for visualizing muscles, tendons, ligaments, and organs.
- Multi-Planar Imaging: MRI can generate images in various planes (sagittal, coronal, axial) without moving the patient.
- High-Quality Images: MRI produces detailed images with high resolution, aiding in accurate diagnosis.

Limitations

- MRI can be noisy and may require patients to remain still for an extended period during the scan.
- Certain medical conditions or implants, such as pacemakers, may restrict MRI use.
- Claustrophobia can be a concern for some patients due to the confined space of the MRI machine.


R
L
A

8.2CT

Computerized Tomograpy (CT)

- **CT Scan Overview:**
  - Computed Tomography (CT) scan is a medical imaging method that uses X-rays and computer processing to create cross-sectional images of the body.
  - CT scans are faster, making them suitable for emergencies.
  - CT scans use X-rays, which involve minimal radiation exposure.
  - Better for bone imaging and detecting internal bleeding.
- **How CT Scans Work:**
  - CT scanners rotate around the body, taking X-ray images from different angles.
  - Computer combines these images to create detailed cross-sectional slices
- **Clinical Applications:**
  - Commonly used for detecting injuries, tumors, infections, and fractures.
  - Aids in planning surgeries and evaluating treatment responses.
- **Contrast Enhancement:**
  - Contrast dye may be injected to highlight certain structures for better visibility in the images.

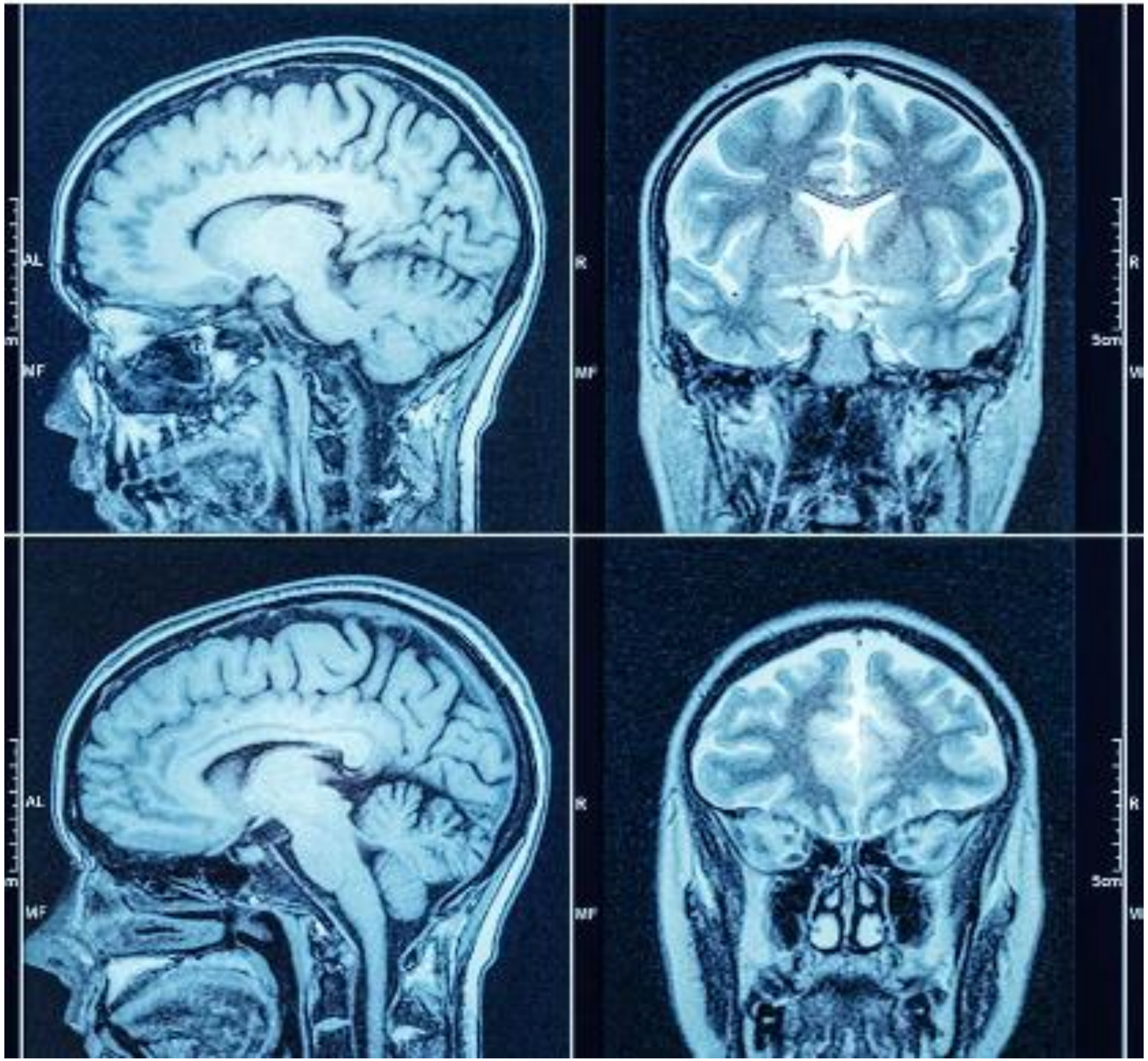
Pros and cons

Advantages

- Provides clear images of bones, tissues, and blood vessels.
- Useful for diagnosing and guiding treatments for various conditions.

Limitations

- Involves radiation exposure, although newer machines use lower doses.
- Some people might be allergic to contrast dye.
- Less detailed images of soft tissues compared to MRI.


AL
MF
R
MF

8.3 Differences

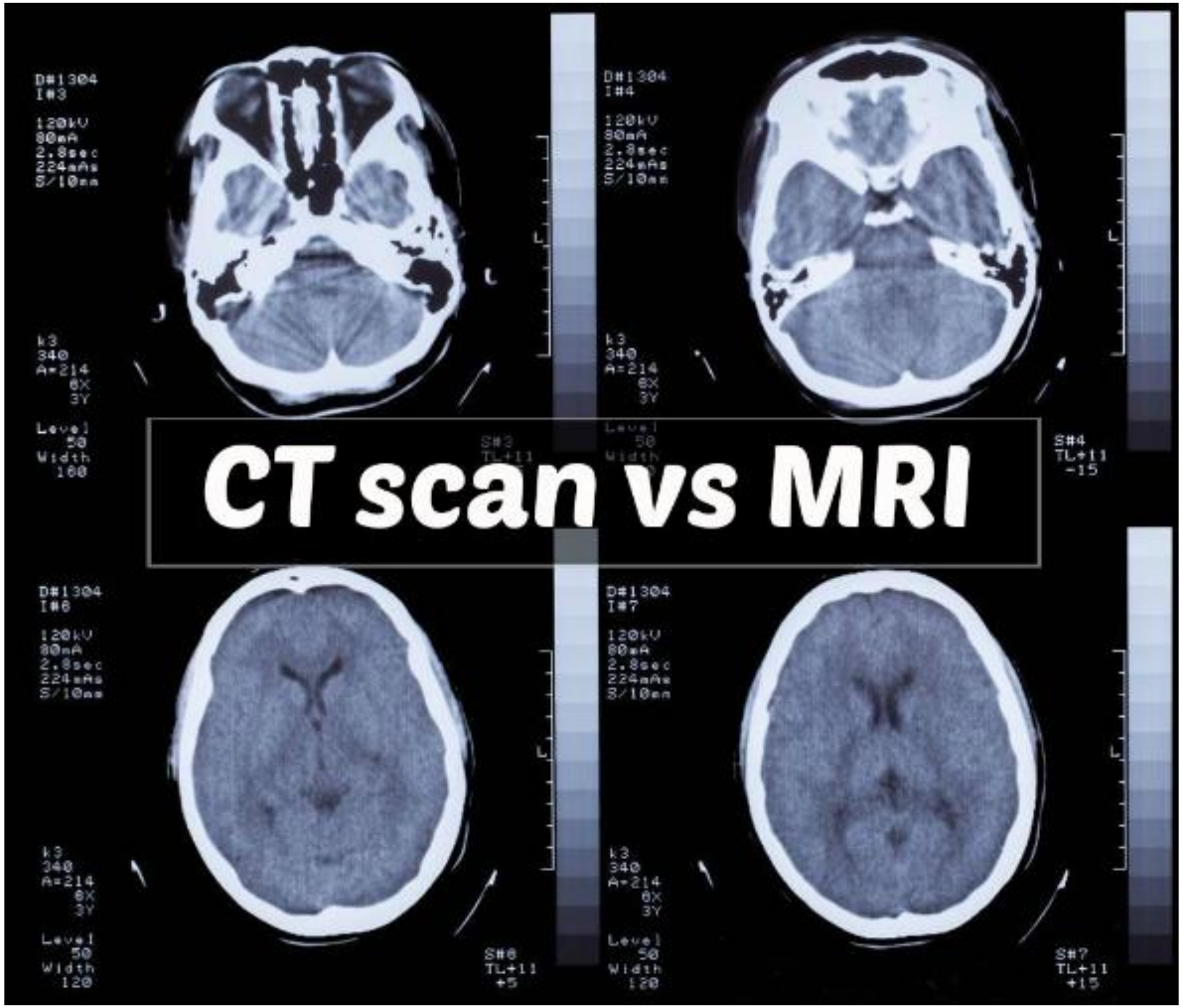
Differences between MRI & CT

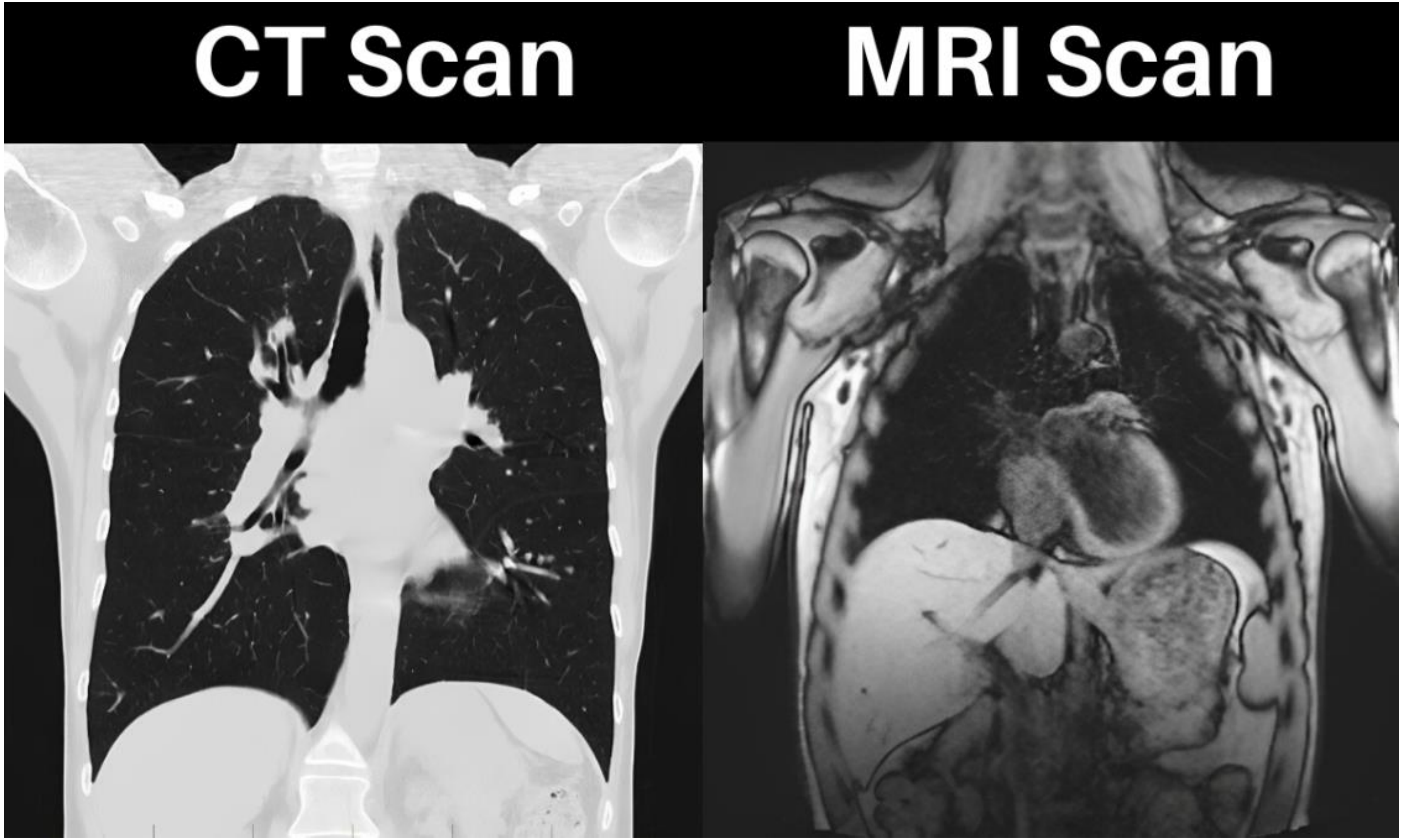
Magnetic Resonance Imaging (MRI):

- **Principle**: Uses magnetic fields and radio waves to create detailed images of soft tissues and organs.
- **Radiation**: Does not involve ionizing radiation; considered safe for multiple scans.
- **Best For**: Visualizing brain, nerves, muscles, ligaments, tendons, and organs.
- **Contrast**: Uses contrast agents to enhance certain areas for better visibility.
- **Image Quality**: Offers exceptional soft tissue contrast, ideal for detailed structural analysis.
- **Time**: Scans take longer; patient needs to remain still during the process.
- **Suitable For**: Evaluating brain disorders, soft tissue injuries, and joint problems

Computed Tomography (CT) Scan:

- **Principle**: Uses X-rays and computer processing to create cross-sectional images.
- **Radiation**: Involves ionizing radiation, but newer machines use lower doses.
- **Best For**: Visualizing bones, blood vessels, and dense tissues.
- **Contrast**: Contrast dye may be used to highlight specific structures.
- **Image Quality**: Provides detailed images of bones, making it excellent for fracture detection.
- **Time**: Scans are quicker; often used in emergency situations.
- **Suitable For**: Detecting trauma, internal bleeding, and bone abnormalities.


CT scan vs MRI


CT Scan
MRI Scan

8.4 Ultrasound

Ultrasound Imaging

- **Principle:** Uses high-frequency sound waves to create real-time images of the inside of the body.
- **Non-Invasive:** Does not involve radiation; considered safe for various applications.
- **Transducer:** A handheld device sends and receives sound waves; also called a probe.
- **Sound Waves:** Bounce off tissues and create echoes, which are converted into images.
- **Clinical Applications:**
- **Pregnancy:** Widely known for monitoring fetal development and health during pregnancy.
- **Diagnostic Imaging:** Used to visualize organs like the heart, liver, kidneys, and uterus.
- **Guidance:** Helps guide medical procedures like biopsies and injections.
- **Vascular Imaging:** Examines blood flow and detects blockages or abnormalities.

Pros and cons

Advantages

- **Real-Time Imaging**: Provides dynamic images as it's happening, allowing for instant feedback.
- **Non-Radiation**: Safe for repeated use, making it suitable for prenatal monitoring.
- **Portable**: Portable devices are available for bedside or point-of-care imaging.
- **Affordable**: Generally more cost-effective than other imaging methods.

Limitations

- **Limited Penetration**: Sound waves may not penetrate dense tissues or bones well.
- **Operator Dependence**: The quality of images can vary based on the operator's skill.
- **Limited Details**: Provides less detailed images compared to MRI or CT scans.
- **Patient Factors**: Limited use in obese patients or those with gas-filled organs


11.08.1984
RAB4-8-D/OB
MI 1.1
Dr. Moroder ecofetale.com
GA=12w3d
8.3cm/1.4/16Hz
TIs 0.1
02.02.2012
12:41:36
Voluson
E8
Routine
Har-high
97
Gn 10
C6 / M7
P5 / E3
SRI II 5
CRL
CRL 6.51cm
GA 12w6d 71.8%

9. Medical Products and Companies

Medicals Products and Companies

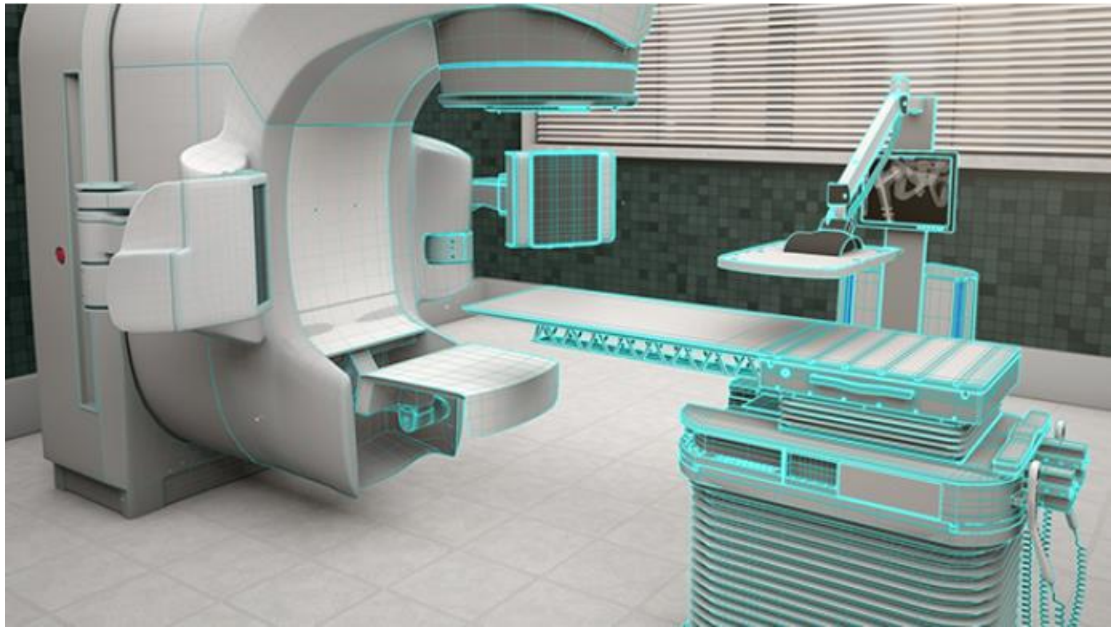
- **Medical Products:**
  - Siemens PLM offers solutions for medical devices and pharmaceuticals.
  - Olympus Medical provides endoscopic surgical products like POWERSEAL.
- **Key Features:**
  - Innovation and precision are hallmarks of medical products.
  - Compliance with rigorous regulatory standards ensures safety.

Medicals Comapnies and Impact

- **Medical Companies:**
  - Siemens PLM: Enables efficient medical device and pharmaceutical manufacturing.
  - Olympus Medical: Global leader in endoscopy and minimally invasive therapy.
- **Importance:**
  - Medical products are vital for diagnosis, treatment, and monitoring.
  - Ongoing research and development lead to medical advancements.
- **Innovation and Impact:**
  - Technological advances enhance accuracy and patient experience.
  - Medical companies have a global reach, positively impacting healthcare systems worldwide.


OLYMPUS


Operational Excellence
Operate digitally today to excel tomorrow



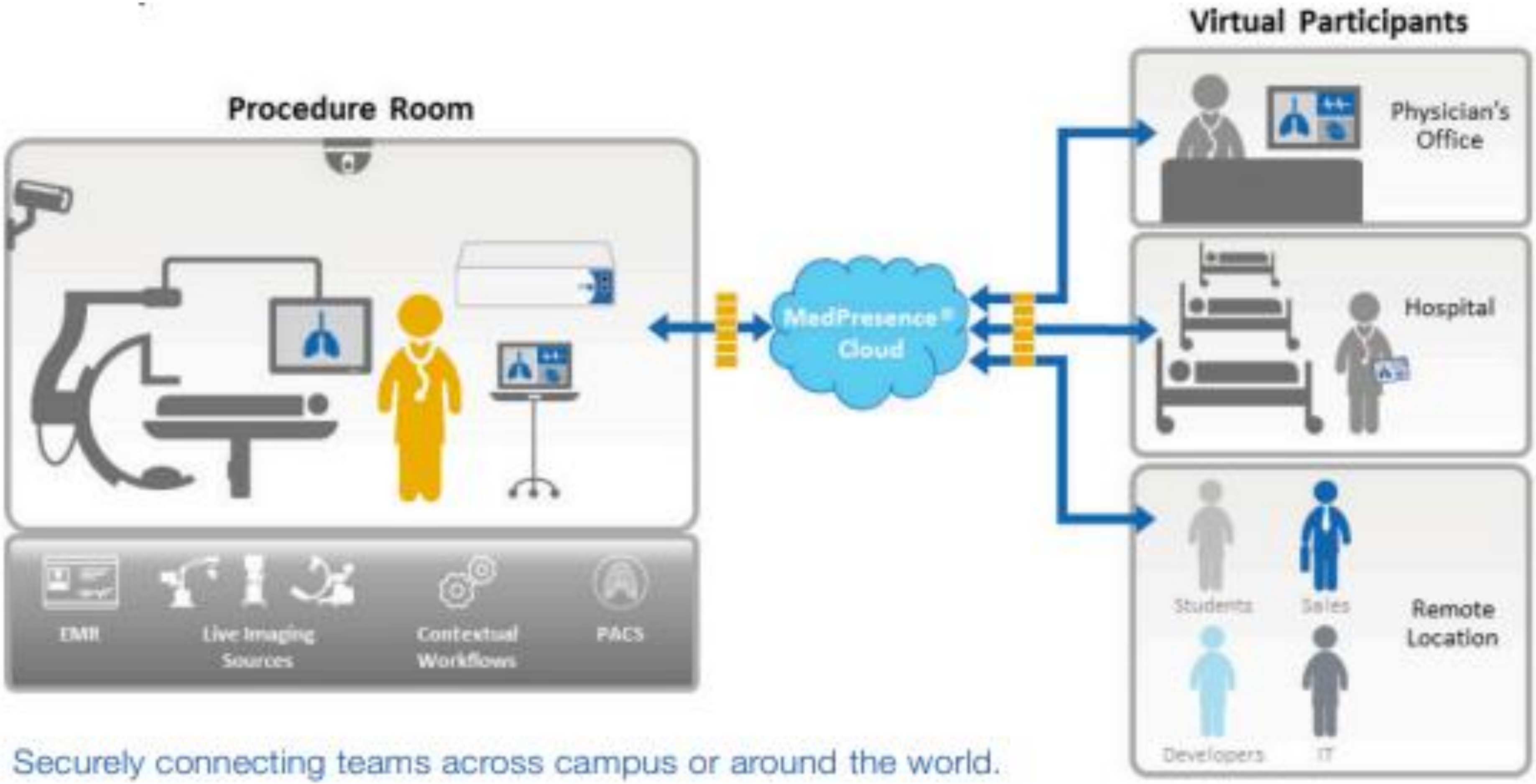
9.1 Medpresence

Olympus MedPresence System Overview

- **MedPresence System:**
  - Telemedicine solution by Olympus.
  - Enables real-time remote collaboration in healthcare.
- **Features:**
  - Live video sharing for medical procedures.
  - Secure and compliant with data protection standards.
  - Interactive tools for discussion and annotations.
- **Benefits:**
  - Remote learning and training for medical professionals.
  - Expert consultations during surgeries.
  - Global networking for enhanced patient care.

Applications and Impact of MedPresence

- **Applications:**
  - Enhances medical education through virtual observation.
  - Enables expert guidance during surgeries.
  - Facilitates telemedicine consultations.
- **Impact:**
  - Increases efficiency by reducing travel for medical professionals.
  - Fosters collaboration among global healthcare experts.



Securely connecting teams across campus or around the world.