Real time system

Lecture 6 : Real-time operating systems (RTOS)

ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.

Introduction: what is the OS?

- The Operating System OS is a program that acts as an intermediary between the user of a computer and the computer hardware.

- The purpose of the OS is to provide an environment in which a user can execute programs in a convenient and efficient manner.

Introduction: what is RTOS?

- Real-time Operating System RTOS is a special purpose OS.

1. Operating systems operate the digital processor with peripherals.

2. The system has **certain time constraints**.

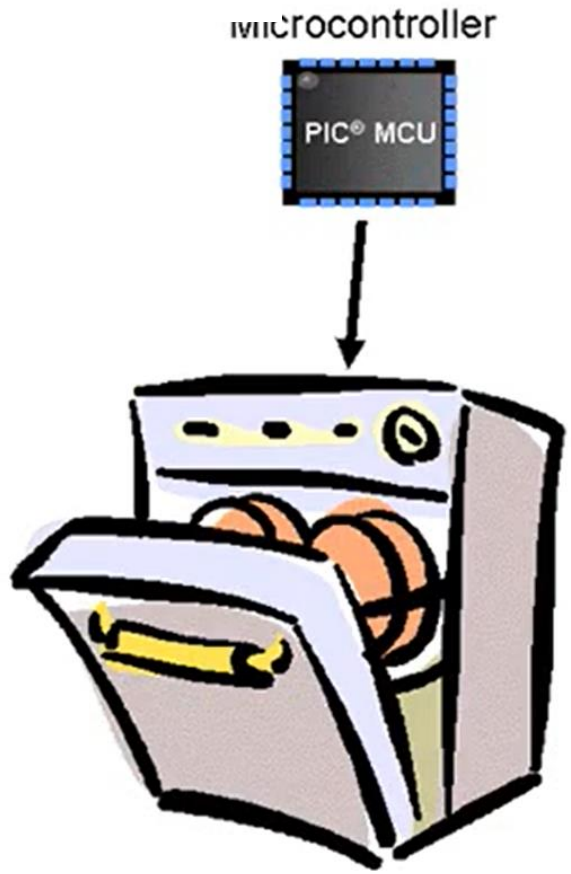
If both conditions **1 & 2 are applied** then we will have **RTOS**.

Examples of RTOS?

- Any OS that controls:

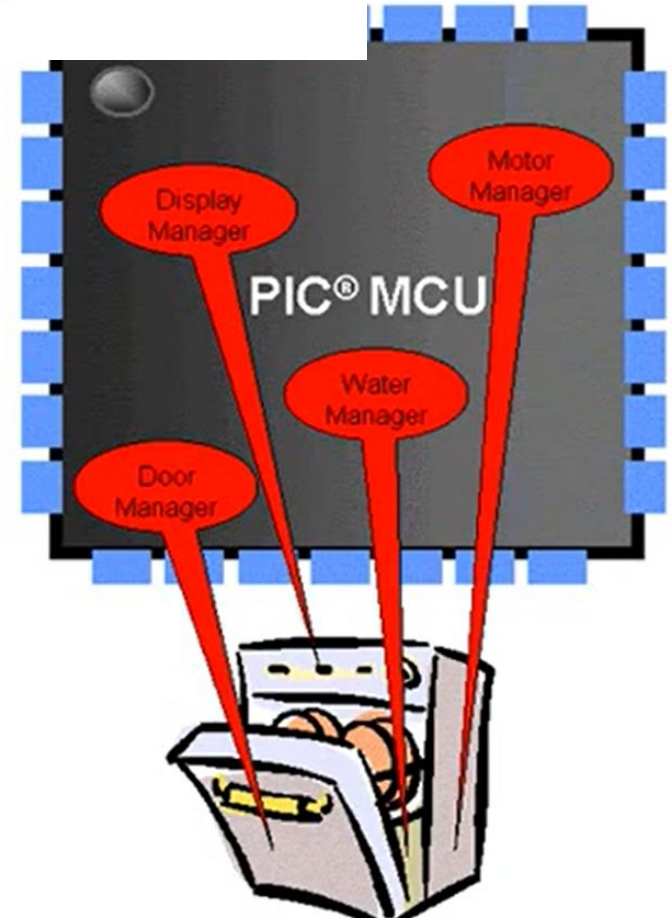
1. Medical imaging systems.
2. Flight control systems
3. Automobile-engine fuel injection.
4. Home-appliance controllers
5. Virtual reality.

System



- A dishwasher is a system
  - Water intake/drain
  - Motor operation
  - Door opening/closing
  - Display status
- A microcontroller manages system operations

Task



- Group of instructions performing a function of a system
- Dishwasher system may have following tasks:
  - Water Manager
  - Motor Manager
  - Door Manager
  - Display Manager

Multi-tasking

- Doing multiple things at once
- For a dishwasher system, the embedded microcontroller may perform the following tasks (all at once):
  - Water Management
  - Motor Management
  - Respond to Door Opening/Closing
  - Display Various Status Information

Deadline

- Essential aspect of real-time system
- Tasks perform their function within prescribed deadline
  - Failure may result in severe consequences
- Deadlines can be
  - Hard
  - Soft

Hard RTOS specifications

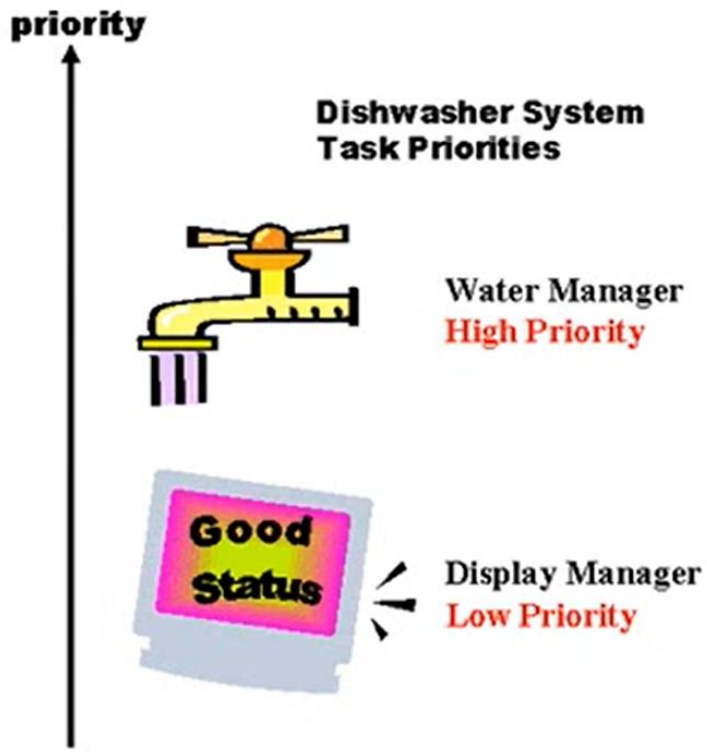
- Critical (or Real-time) tasks (processes) must be completed on time.
- Secondary storage are limited or missing.
- Short-term memories are used instead.

Soft RTOS Specifications

- A critical (or Real-time) tasks (processes) gets priority over other tasks and retain that priority until it completes.

- Kernel delays must be bounded because the RT task cannot wait indefinitely for the kernel to run it.

Priority

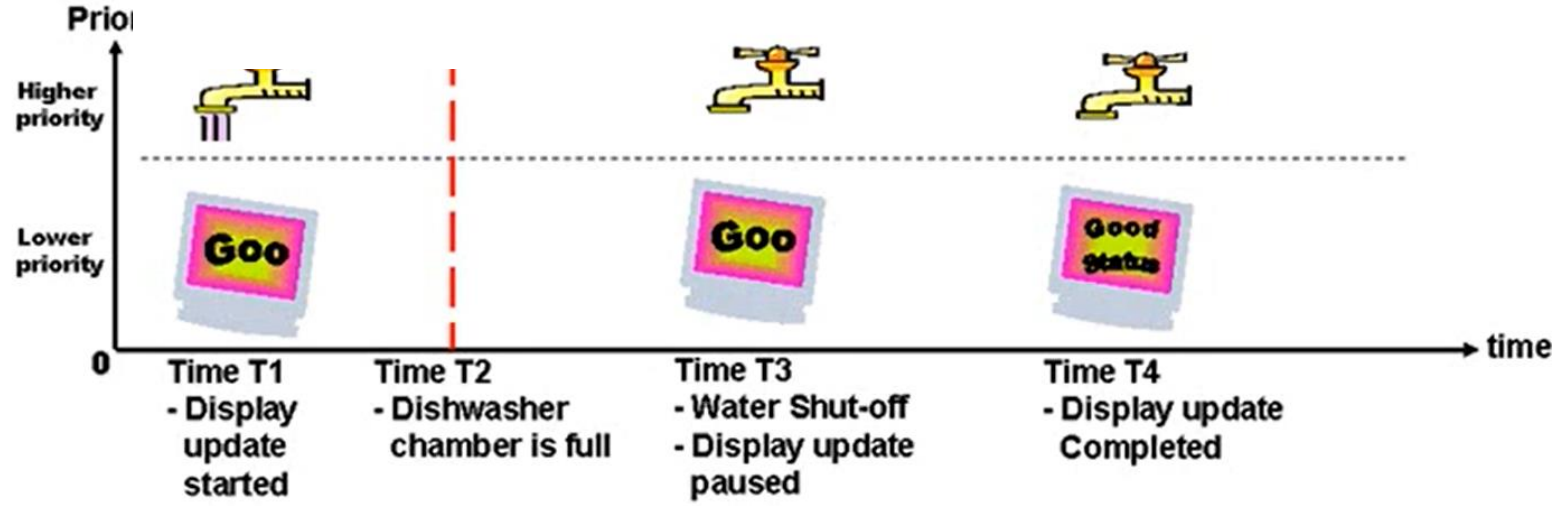


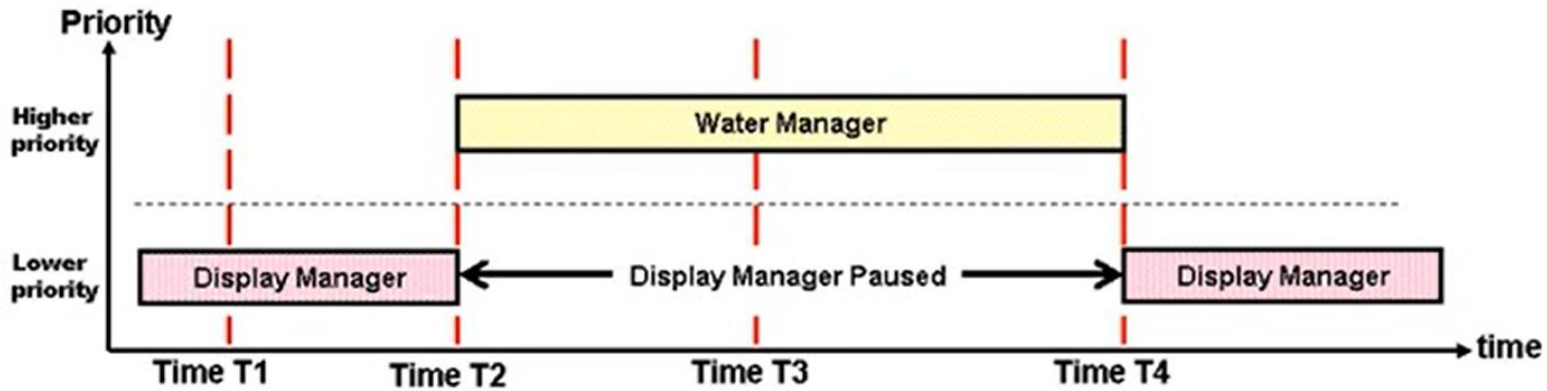
- Every task in a system has priority
- Priority determines the relative order of the importance of tasks in the system
  - Important tasks have higher priority
  - Less important tasks have lower priority

Preemption

- Prerogative of higher priority tasks in a system
- Allows higher priority task to pause a lower priority task
- Necessary to maintain system integrity of a real-time system

Preemption Example





RTOS Constraints

The following types of constraints are considered:

1. Timing constraints: meet your deadline.
2. Resource constraints: access only available resources.

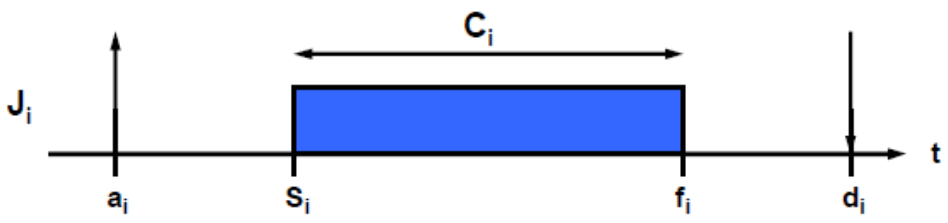
1. Timing Constraints

- Real-time systems are characterized mostly by timing constraints
- Typical timing constraint: *deadline*

  deadline = time before which task has to be performed
- Deadline missing separates two classes of RT systems:
  - Hard : missing of deadline can cause catastrophic consequences
  - Soft : missing of deadline decreases performance of system

Parameters to Characterize RT-task Ji

- Arrival time ai : the time Ji becomes ready for execution also called *request time or release time, denoted by ri*
- Computation time Ci: time necessary for execution without interruption
- Deadline di: time before which task has to be completed its execution
- Start time Si: time at w
- Finishing time fi: time



Typical parameters of a real-time task

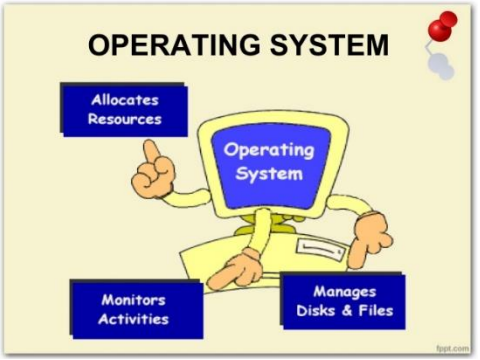
2. Resource Constraints

- **Process‘s view:**

  Resource is any SW structure to be used by process

  Examples: data structure, set of variables, memory area, file or a device driver (Hardware). So, resources might be:
- **Private resource** : dedicated to a particular process
- **Shared resource** : to be used by more than any process
- **Exclusive resource** : shared resource where simultaneous access from different processes is not allowed.

Main Components of RTOS



- **Timing Services**: Clocks, Timers, etc.
- **Scheduler**: Choose the next process to run
- **Communication Mechanism**: Semaphores, Message Queues, Queues, etc.
- **Power Management**: For low power devices, power management is generally part of the RTOS since it knows the state of the device.
- **Resource Management:** Allocates memory and processor resources.
- **Peripheral Drivers**: UART, SPI, I2C, etc.

Scheduler

Scheduling disciplines are algorithms used for distributing resources among parties which simultaneously and asynchronously request them.

Scheduling disciplines are used in routers (to handle packet traffic) as well as in operating systems (to share CPU time among both threads and processes), disk drives (I/O scheduling), printers (print spooler), most embedded systems, etc.

Summary

- Basic unit of a real-time system is a Task
- There may be multiple tasks in a system
- Tasks have deadlines
- Tasks have priority
- Higher priority task preempt lower priority tasks
- Scheduler enforces priority based preemption

Real time system
Lecture 1: introduction

ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.

Outline

- Definition system .
- classic representation of RTS .
- Definition response time .
- Real time system
- Embedded system
- Type of real time system .
- Real –time punctuality
- Event and release.

- Definition: System

- A system is a mapping of a set of inputs into a set of outputs.

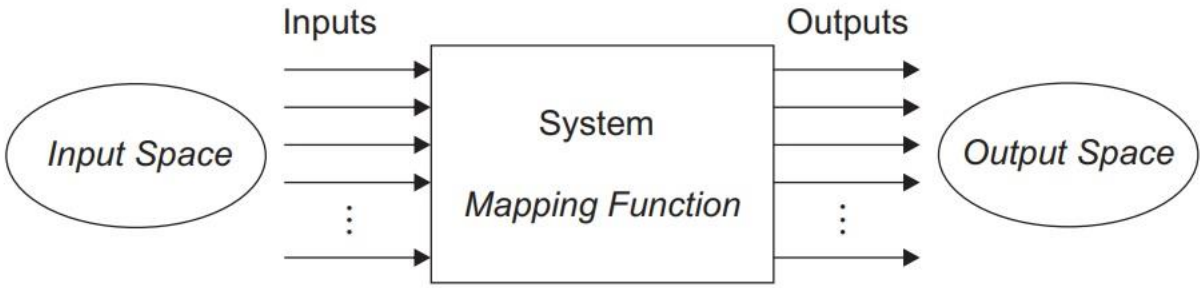


Figure 1.1. A general system with inputs and outputs.

1. A system is an assembly of components connected together in an organized way.
2. A system is fundamentally altered if a component joins or leaves it.
3. It has a purpose.
4. It has a degree of permanence.
5. It has been defined as being of particular interest.

- **Example : A Real Time control system**

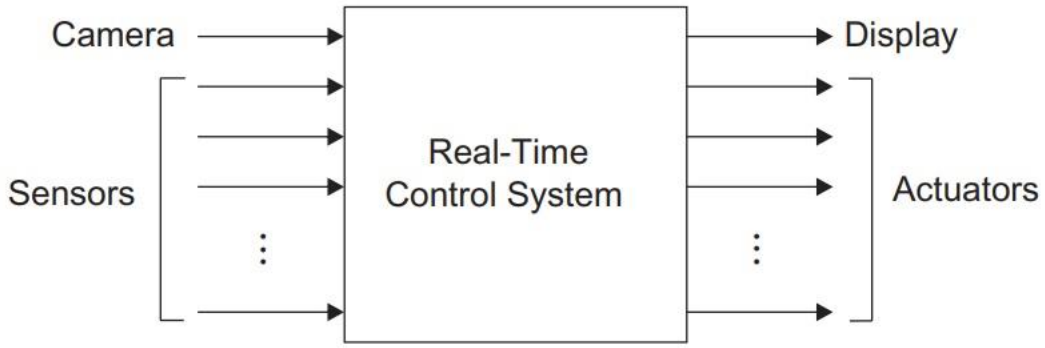


Figure 1.2. A real-time control system including inputs from a camera and multiple sensors, as well as outputs to a display and multiple actuators.

1- input are **excitation** and outputs are corresponding **responses.**
2. **Input** and output may be **digital or analog** .
3. **Input** are associated with sensor ,cameras ,etc.
4. **Output** with actuator ,display ,etc.

Classic representation of RTS

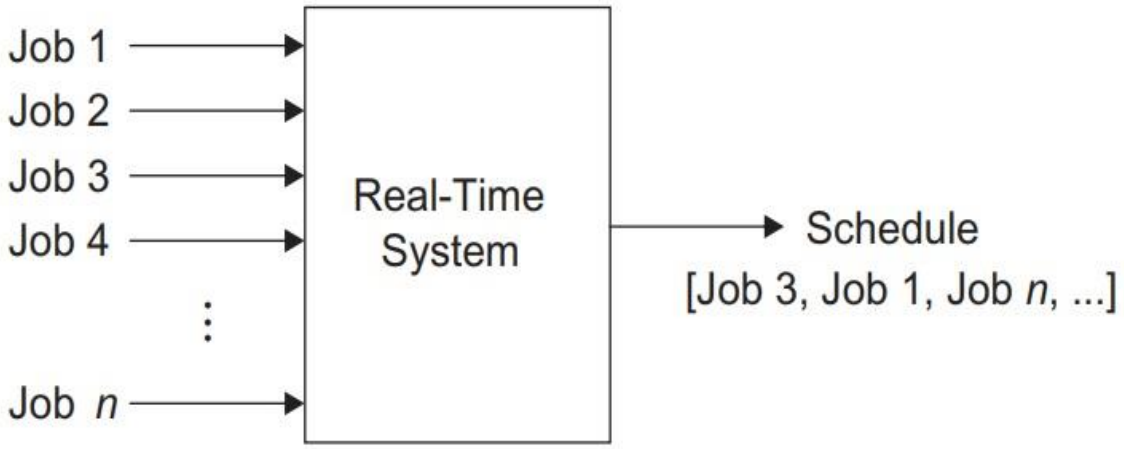


Figure 1.3. A classic representation of a real-time system as a sequence of schedulable jobs.

- A sequence of jobs to be scheduled and performance to be predicted .
- Ignores the usual fact that the input source and hardware under control may be high complex.

Definition: Response Time

The time between the presentation of a set of inputs to a system and the realization of the required behavior, including the availability of all associated outputs, is called **the response time of the system.**

- How fast and punctual dose it need to be ?
  - depended on the specific real time system..
- but what is the real time system ?
- The previous definitions set the stage for a practical definition of a real-time system.

Definition: Real-Time System

- A real-time system is a computer system that must satisfy bounded response time constraints or risk severe consequences, including failure.
- A real-time system is one whose logical correctness is based on both the **correctness** of the outputs and their **timeliness**.

But what is a “failed” system? In the case of the space shuttle or a nuclear power plant, for example, it is painfully obvious when a failure has occurred. For other systems, such as an automatic bank teller machine, the notion of failure is less obvious.

Definition: Failed System

- A failed system is a system that cannot satisfy one or more of the requirements stipulated in the system requirements specification..

- Because of this definition of failure, rigorous specification of the system operating criteria, including timing constraints is necessary.

Definition: Embedded System

An embedded system is a system containing one or more computers (or processors) having a central role in the functionality of the system, but the system is not explicitly called a computer.

- for example, a modern automobile contains many embedded processors the control airbag deployment, antilock braking, air conditioning, fuel injection.
- A real time system may be **embedded system or non embedded** . But it always **reactive..**

degree of RTS (Soft ,Hard and Firm real time system)

Soft Real-Time System

A soft real-time system is one in which performance is degraded but not destroyed by failure to meet response-time constraints.

Hard Real-Time System

A hard real-time system is one in which failure to meet even a single deadline may lead to complete or catastrophic system failure.

Firm Real-Time System

A firm real-time system is one in which a few missed deadlines will not lead to total failure, but missing more than a few may lead to complete or catastrophic system failure

Soft ,Hard and Firm real time system

System	Real-Time Classification	Explanation
Avionics weapons delivery system in which pressing a button launches an air-to-air missile	Hard	Missing the deadline to launch the missile within a specified time after pressing the button may cause the target to be missed, which will result in catastrophe
Navigation controller for an autonomous weed-killer robot	Firm	Missing a few navigation deadlines causes the robot to veer out from a planned path and damage some crops
Console hockey game	Soft	Missing even several deadlines will only degrade performance

Where do deadline come form?

- Deadlines are based on the underlying physical phenomena of the system under control .

- **Punctuality** is another measure related to response times ..

- particularly important in periodically sampled systems with high sampling rate (e.g , in audio and video signal processing)

Where do deadline come form?

Real-time punctuality means that every response time has an average value, t_R, with upper and lower bounds of $t_R + \varepsilon_U$ and $t_R - \varepsilon_L$, respectively, and $\varepsilon_U, \varepsilon_L \to 0^+$

- In all practical systems, the values of ε_U and ε_L are **nonzero**, though they may be *very small*.
- The nonzero values are due to cumulative latency & propagation-delay components *(hardware / software)*.
- Such response times contain **jitter** within the interval $t \in [-\varepsilon_L, +\varepsilon_U]$ *(jitter is the amount of variation in response time, usually measured in ms "milliseconds").*

example : Where response time come form?

- An elevator door is automatically operated and it may have a capacitive* safety edge for sensing possible passengers between the closing door blades.

- Thus, the door blades can be quickly reopened before they touch the passenger and cause discomfort or even threaten the passenger's safety.

- What is the required system response time from when it recognizes that a passenger is between the closing door blades and starting to reopen the door?

Door Reopening Example (Cont'd)

This response time consists of five independent latency components:

Sensor:	$t_{\text{SE\_min}} = 5\ \text{ms}$	$t_{\text{SE\_max}} = 15\ \text{ms}$
Hardware:	$t_{\text{HW\_min}} = 1\ \mu\text{s}$	$t_{\text{HW\_max}} = 2\ \mu\text{s}$
System software:	$t_{\text{SS\_min}} = 16\ \mu\text{s}$	$t_{\text{SS\_max}} = 48\ \mu\text{s}$
Application software:	$t_{\text{AS\_min}} = 0.5\ \mu\text{s}$	$t_{\text{AS\_max}} = 0.5\ \mu\text{s}$
Door drive:	$t_{\text{DD\_min}} = 300\ \text{ms}$	$t_{\text{DD\_max}} = 500\ \text{ms}$

Now, we can calculate the **minimum and maximum values of the composite response time:** $\boldsymbol{t_{\text{min}} \approx 305\ ms,\ t_{\text{max}} \approx 515\ \text{ms}}$

- The *overall response time is dominated by the door drive's response time* containing the deceleration time of the moving door blades.

Event and release time

Definition: Event

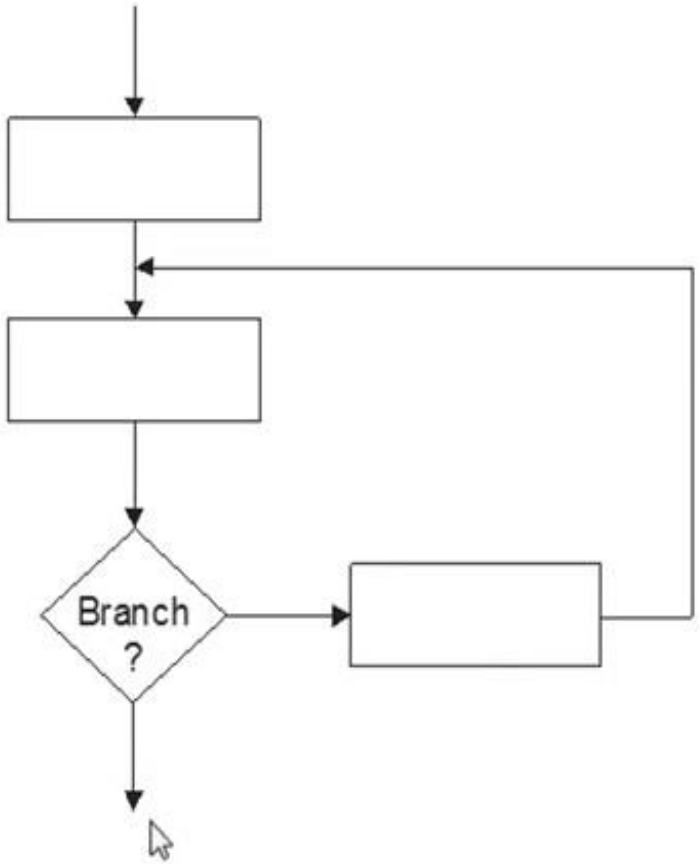
- Any occurrence that causes the program counter to change non-sequentially is considered a change of flow-of-control, and thus an event.

Definition: Release Time

- The release time is the time at which an instance of a scheduled task is ready to run, and is generally associated with an interrupt.

Change in Flow of Control

- A change in state results in a change in the flow-of-control.
- The decision block suggests that the program flow can take alternative paths (*see right*).
- `case`, `if-then`, and `while` statements represent a possible change in flow-of-control.
- Invocation of procedures in Ada and C represent changes in flow-of-control.
- In C++ and Java, instantiation of an object or the invocation of a method causes the change in flow-of-control.



Taxonomy of Events

- An event can be either synchronous or asynchronous:
  - *Synchronous* events are those that occur at predictable times in the flow-of-control.
  - *Asynchronous* events occur at unpredictable points in the flow-of-control and are usually caused by external sources.
- Moreover, events can be periodic, aperiodic or sporadic:
  - A real-time clock that pulses regularly is a *periodic* event.
  - Events that do not occur at regular periods are called *aperiodic*.
  - Aperiodic events that tend to occur very infrequently are called *sporadic*.

Example: Various Types of Events

Type	Periodic	Aperiodic	Sporadic
Synchronous	Cyclic code	Conditional branch	Divide-by-zero (trap) interrupt
Asynchronous	Clock interrupt	Regular, but not fixed-period interrupt	Power-loss alarm

Deterministic system

Definition: Deterministic System

- **A system is deterministic, if for each possible state and each set of inputs, a unique set of outputs and next state of the system can be determined.**
- **Event determinism means the next states and outputs of a system are known for each set of inputs that trigger events. Thus, a system that is deterministic.**

CPU Utilization or Time-Loading Factor

- The final term to be defined is a critical measure of real-time system performance.
- Because the CPU continues to execute instructions as long as power is applied, it will more or less frequently execute instructions that are not related to the fulfillment of a specific deadline.
- The measure of the relative time spent doing *non-idle processing* indicates how much real-time processing is occurring.

Definition: ***CPU Utilization Factor***

The CPU utilization or time-loading factor, *U*, is a relative measure of the non-idle processing taking place.

CPU utilization

- A system is said to be time-overloaded if $U > 100\%$.
- Systems that are too highly utilized are problematic:
  - Additions, changes, or corrections cannot be made to the system without risk of time-overloading.
- On the other hand, systems that are not sufficiently utilized are not necessarily cost-effective:
  - The system was over-engineered and that costs could likely be reduced with less expensive hardware.
- While a utilization of 50% is common for new products, 80% might be acceptable for systems that do not expect growth.
- However, 70% as a target for U is one of the potentially useful results in the theory of real-time systems where tasks are periodic and independent (*more in Chapter 3*).

CPU Utilization Zones (Cont'd)

Utilization %	Zone Type	Typical Application
< 26	Unnecessarily safe	Various
26 – 50	Very safe	Various
51 – 68	Safe	Various
69	Theoretical limit	Embedded systems
70 – 82	Questionable	Embedded systems
83 – 99	Dangerous	Embedded systems
100	Critical	Marginally stressed system
> 100	Overloaded	Stressed system

Calculation of *U*

- *U* is calculated by summing the contribution of utilization factors for each task ..
- Suppose a system has $n \geq 1$ periodic tasks, each with an execution period of p_i .. If task *i* is known to have a *worst-case* execution time of e_i, then the utilization factor, u_i, for task *i* is

$$\boldsymbol{u_i = e_i/p_i} \qquad \textbf{(1.1)}$$

Furthermore, the overall system utilization factor is

$$\boldsymbol{U = \sum_{i=1}^{n} u_i = \sum_{i=1}^{n} e_i/p_i} \qquad \textbf{(1.2)}$$

- In practice, the determination of e_i can be difficult, in which case estimation or measuring must be used.
- For aperiodic and sporadic tasks, u_i is calculated by assuming a worst-case execution period.

Example: Calculation of the CPU Utilization Factor

- An individual elevator controller in a bank of high-rise elevators has the following software tasks with execution periods of pi and worst-case execution times of ei, i ∈ {1, 2, 3, 4}:
- Task 1: Communicate with the group dispatcher (19.2 K bit/s data rate
- and a proprietary communications protocol); p1 = 500 ms, e1 = 17 ms.
- Task 2: update the car position information and manage floor-to-floor
- runs, as well as door control; p2 = 25 ms, e2 = 4 ms.
- Task 3: Register and cancel car calls; p3 = 75 ms, e3 = 1 ms.
- Task 4: Miscellaneous system supervisions; p4 = 200 ms, e4 = 20 ms.

- What is the overall CPu utilization factor?

$$U = \sum_{i=1}^{4} e_i / p_i = \frac{17}{500} + \frac{4}{25} + \frac{1}{75} + \frac{20}{200} \approx 0.31$$

Hence, the utilization percentage is 31%, which belongs to the "very safe" zone of Table 1.3.

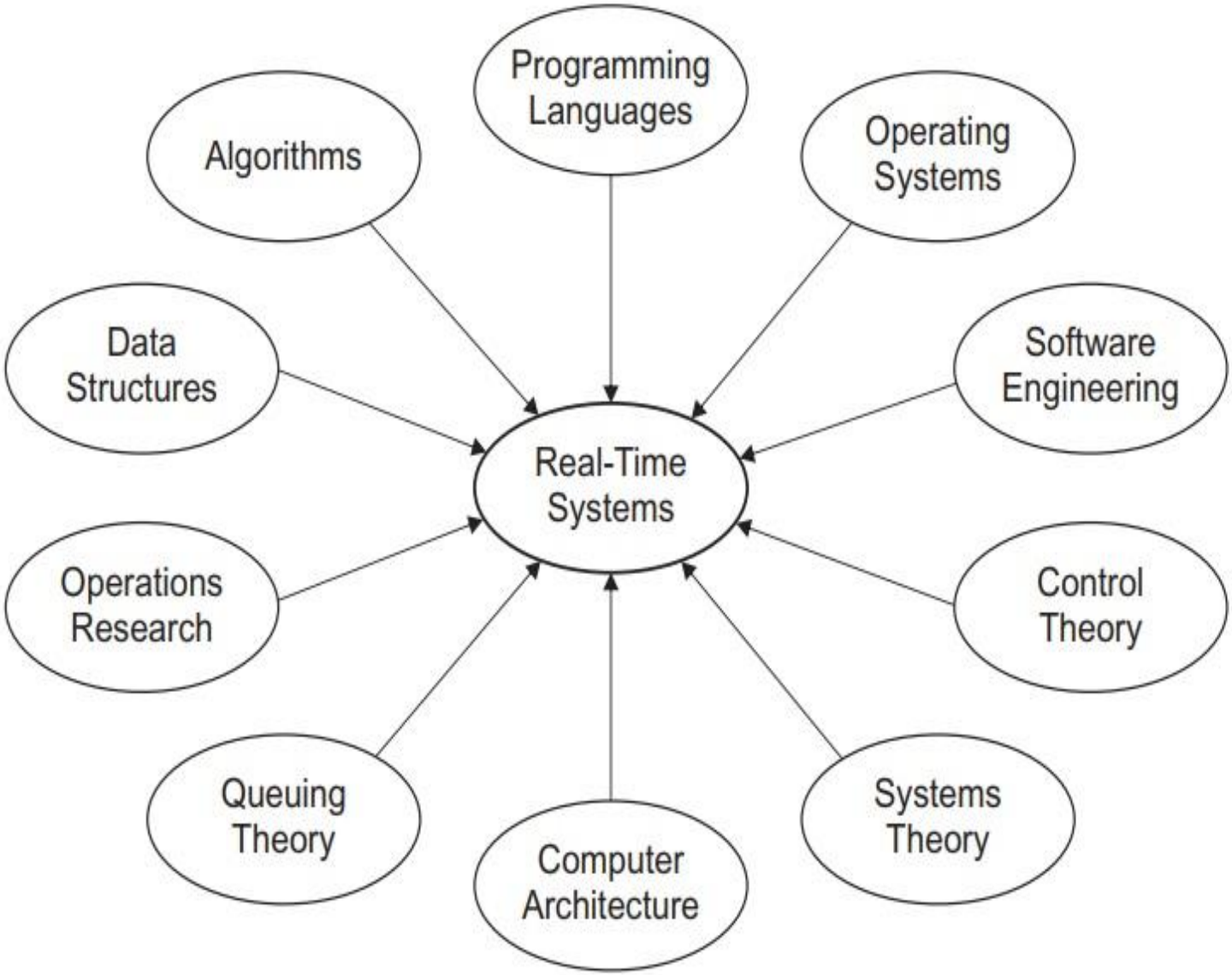


Figure . . A variety of disciplines that affect real-time systems engineering.

Real time system
Lecture 2: sensors

ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.

Theory part

- Introduction
- Sensors
- Signal conditioners
- Microcontroller….
- Data Acquisition and Computer buses
- Embedded Systems
- Real-time OS and Scheduling
- Real-time Scheduling Algorithm
- Embedded System Programming

What is a Real-Time System?

Real-time systems have been defined as: "those systems in which the correctness of the system depends not only on the logical result of the computation, but also on the time at which the results are produced";

- **Examples of temporal constraints**
  - **Few milliseconds** for radar systems.
  - **One second** for machine-man interfaces (in an aircraft for example).
  - **Hours** for some chemical reactions.
  - **24 hours** for weather forecast.
  - **Several months or years** for some ….?

Some Definitions (Timing)

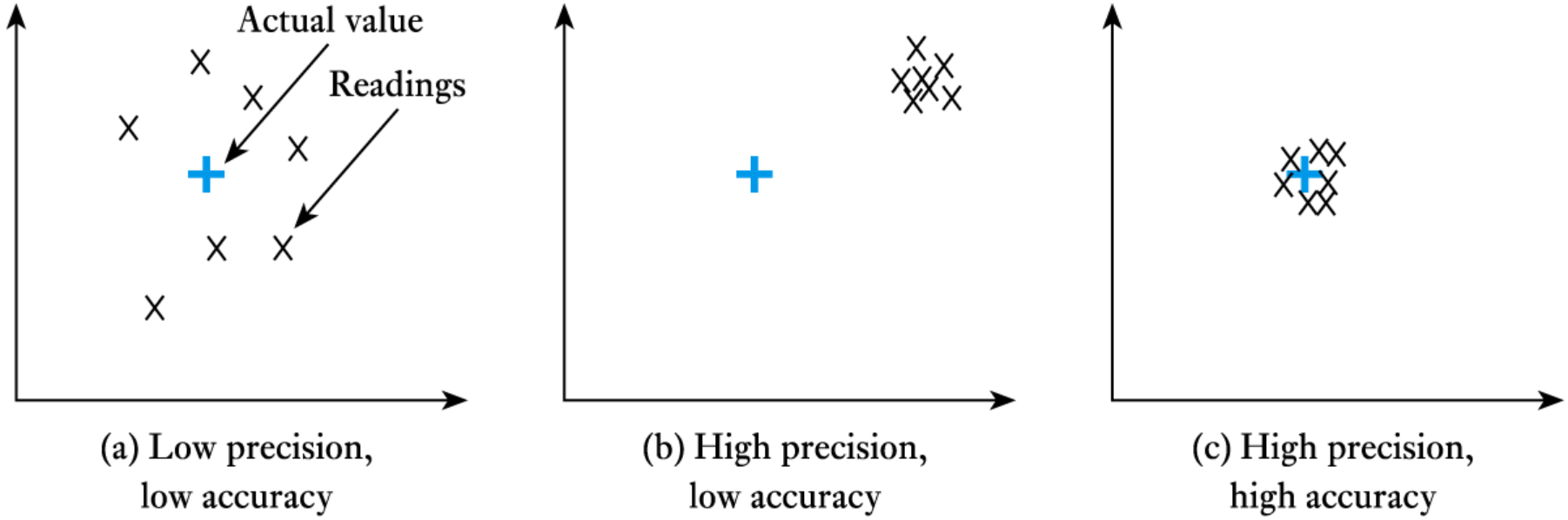
- **Timing constraint:** constraint imposed on timing behavior of a job: hard or soft.
- **Hard real-time:** systems where it is absolutely imperative that responses occur within the required deadline. E.g. Flight control systems.
- **Soft real-time:** systems where deadlines are important but which will still function correctly if deadlines are occasionally missed. E.g. Data acquisition system.

Some Definitions (Measuring)

- **Range:** maximum and minimum values that can be measured
- **Resolution or discrimination:** smallest discernible change in the measured value
- **Error:** difference between the measured and actual values
    - random errors
    - systematic errors
- **Accuracy, inaccuracy, uncertainty:** accuracy is a measure of the maximum expected error
- **Sensitivity:** a measure of the change produced at the output for a given change in the quantity being measured

Some Definitions(Measuring)

- **Precision:** a measure of the lack of random errors (scatter)



(a) Low precision, low accuracy

(b) High precision, low accuracy

(c) High precision, high accuracy

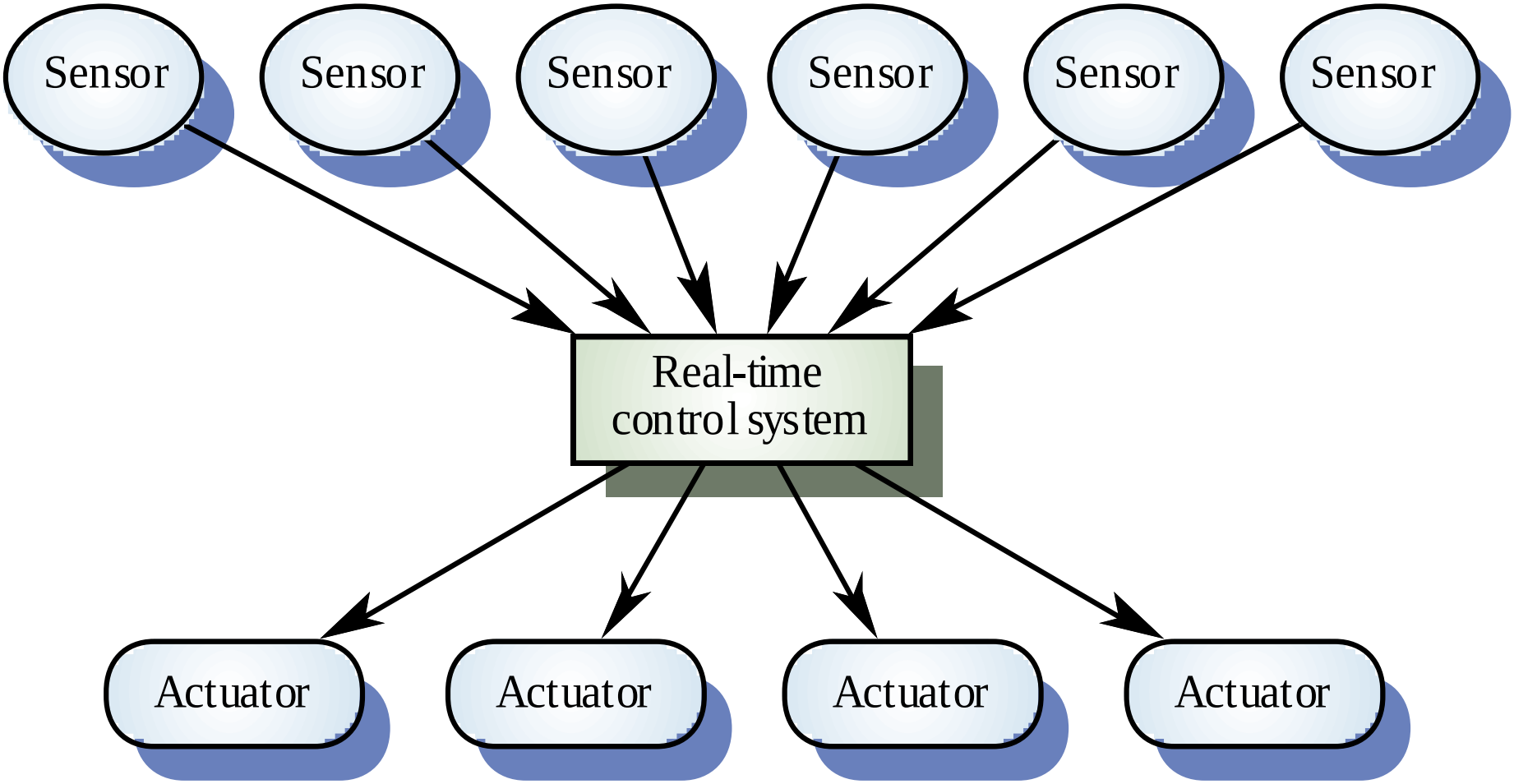
- **Linearity:** maximum deviation from a 'straight-line' response.

Example

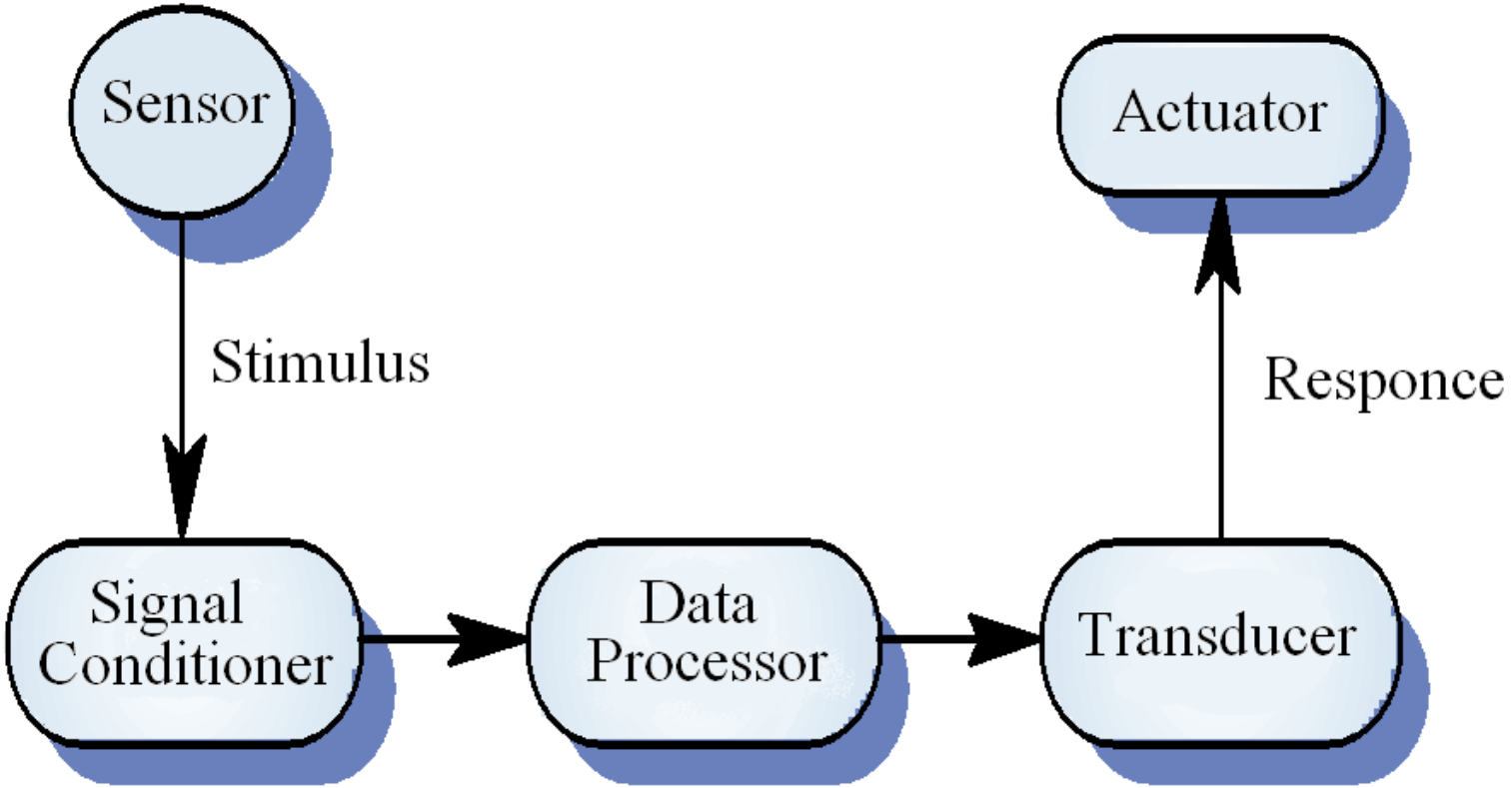
- Calculate the resolution of a Multi-Meter which has a range of ±13V with 8 bit ADC

- 8-bit ADC has 2^8 steps =256
- For a multi-meter of ± 13V the range is 26V
- Step/Resolution is 26/256 =101 mV

A Real-Time System Model



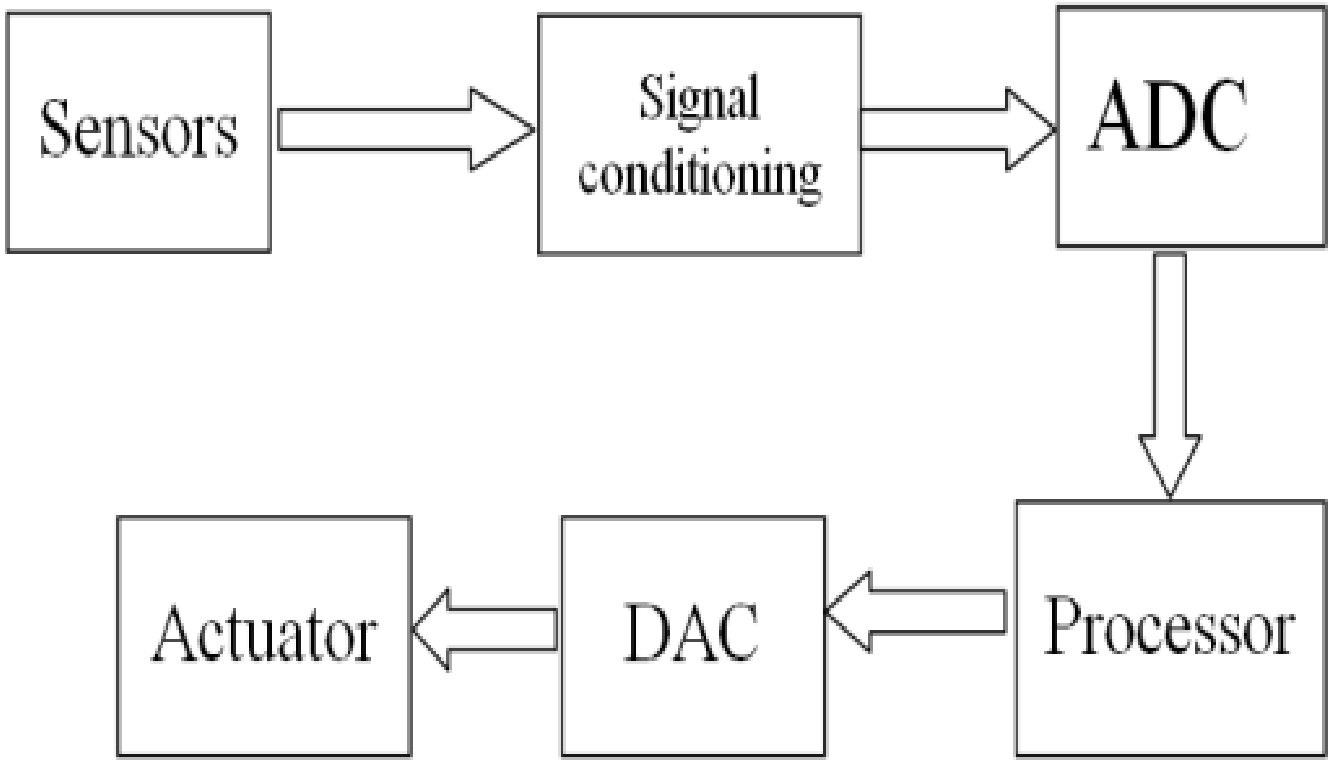
Sensor/Actuator Processes



System Elements

- Sensors control processes
  - Collect information from sensors. May buffer information collected in response to a sensor stimulus
- Data processor
  - Carries out processing of collected information and computes the system response
- Actuator control
  - Generates control signals for the actuator

A Real-Time System Model



Sensors and There Classification

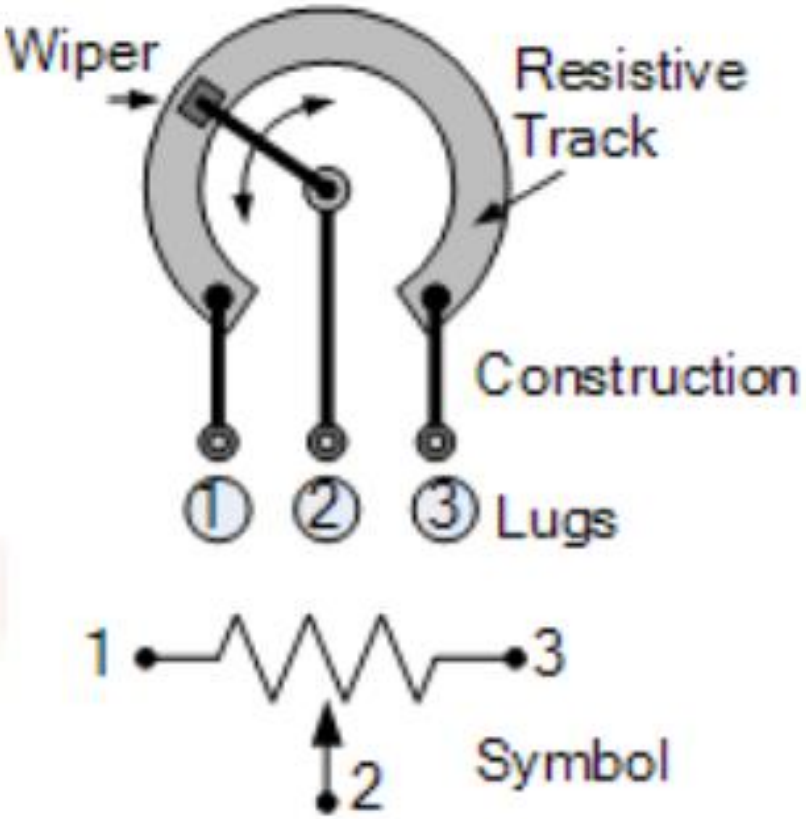
- Sensors and actuators are examples of **transducers**

  A transducer is a device that converts one physical quantity into another

- Almost any physical property of a material that changes in response to some excitation can be used to produce a sensor
  - widely used sensors include those that are:

1. Resistive
2. Capacitive
3. Inductive
4. Electromagnetic
5. Thermoelectric
6. Elastic
7. Piezoelectric
8. Photoelectric
9. Pyroelectric
10. Hall effect

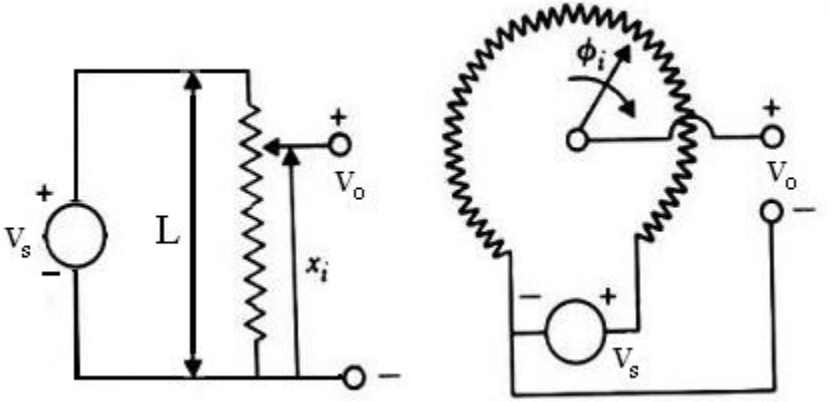
1.Resistive Sensors



1.Resistive Sensors

1.1 Potentiometers: It is a resistance element with a sliding contact which is moved over the length of the element.

- Used for monitoring Linear or circular displacements.



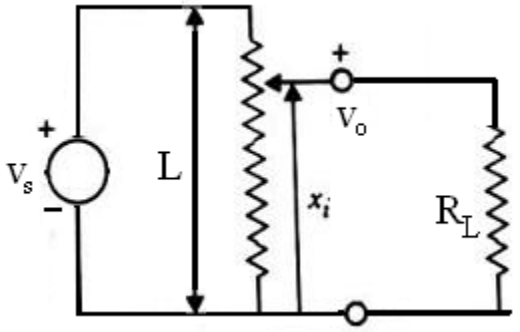
- The fraction ratio (f) is equal to $f = x/L = \phi i/\phi t$

$$V_o = V_s \frac{x}{L} \text{ or } V_s \frac{\phi_i}{\phi_t}$$

- If the total track resistance =Rp then the resistance between the sliding terminal and the reference terminal = f Rp

1.Resistive Sensors

Potentiometers are linear elements (Vo is linearly proportional to Vs) but as a load is placed across the output linearity disappear and error is introduced.



$R_{total} = f R_p R_L / f R_p + R_L$

$V_o = V_s * R_{total} / R_{total} + (1-f) R_p$

$$= V_s \frac{f}{(R_p/R_L) f (1-f) + 1}$$

$$\text{Error} = f\ V_s - V_o = V_s (R_p/R_L)(f^2 - f^3)$$

$d(\text{error})/df = 0 \longrightarrow f = 2/3$ for max. error

1.Resistive Sensors

EX1: A potential resistor of 500W is connected with a multi-meter which has an internal resistor of 10KW calculate the error if| the Vs=4v when f=0.5 (Vo=?)

$$Error = \frac{vsR_p}{R_L} * (f^2 - f^3)$$

$$Error = 4 * \frac{500}{10000} * (0.5^2 - 0.5^3) = 0.025 \ \ (2.5\%)$$

 What happens when RL = 5KW instead of 10KW?

$$Error = \frac{vsR_p}{R_L} * (f^2 - f^3)$$

Good multi-meter should have low or high input impedance?

1.Resistive Sensors

- H .w : If a voltmeter of 10KW internal resistance is connected to a potentiometer of 500W total resistance which is connected to a 10V voltage source
- 1)find the error if the slide is a)at the middle b)at the position for max. error.
- 2)who does reducing the voltage of the source effect the error, what is its disadvantage suggest another way to reduce the error

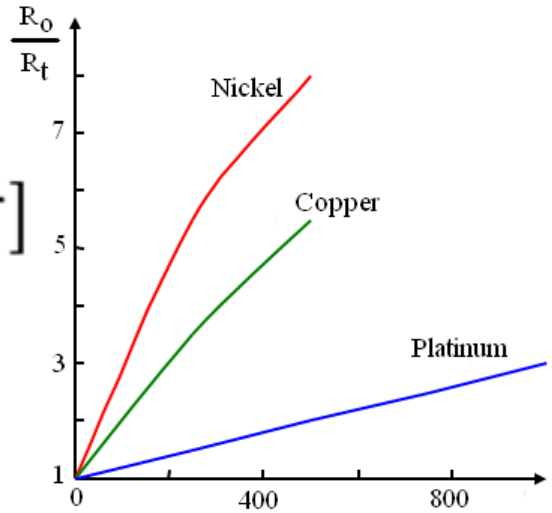
1.Resistive Sensors

1.2 Resistance Temperature Devices (RTDs):

RTDs are made of materials whose resistance changes in accordance with temperature

a)Metals

$$R_T = R_0\left[1+\alpha_1 T+\alpha_2 T^2+\cdots\alpha_n T^n+\right]\cong R_0\left[1+\alpha_1 T\right]$$



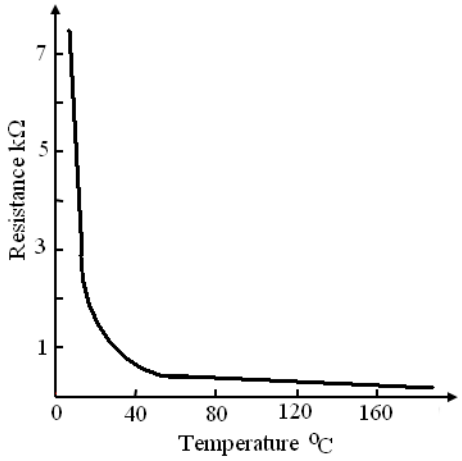
R_T: R at t temperature

R_0: R at 0 C°

α: Temperature coefficient

Why Platinum is used widely?

- **Stable (linear)**
- **Wide range of temperature**



Example

- A platinum resistance thermometer has a resistance of 100 ohm at 0C. Determine the change in resistance that will occur when the temperature rises to 30 C if the temperature coefficient is $0.0039C^{-1}$

$$R_t = R_0 + R_0 \alpha T$$

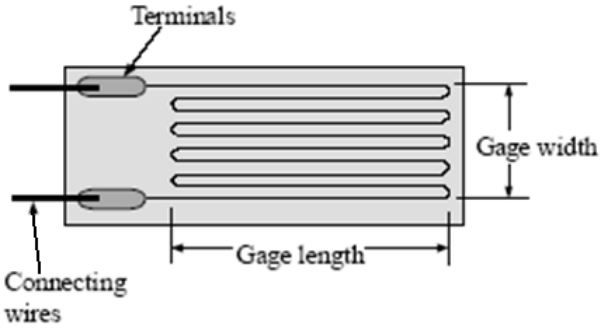
$$\Delta R = R_0 \alpha T = 100 * 0{\cdot}0039 * 30 = 11{\cdot}7\Omega$$

1.Resistive Sensors

1.3 Strain Gauge: consists of a resistance element in the form of a flat coil of wire

Resistance is related to length and area of cross-section of the resistor and resistivity of the material as

$$R \equiv \frac{\rho l}{A}$$



By taking logarithms and differentiating both sides, the equation becomes

$$\frac{dR}{R} = \underbrace{\frac{dl}{l} - \frac{dA}{A}}_{\text{Dimensional}} + \underbrace{\frac{d\rho}{\rho}}_{\text{piezoresistance}} \qquad \text{Where} \quad \frac{dl}{l} = \varepsilon$$

$$\frac{\Delta R}{R} = GE$$

E: Strain

G: sensitivity or gauge factor

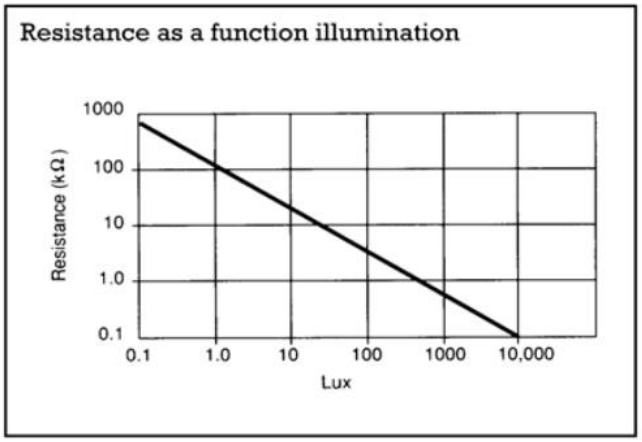
Example

- An electrical strain gauge has a resistance of 120Ω and a gauge factor of 2.1. Find the change in resistance when a strain of 0.0005 is applied along the length.

- $\frac{\Delta R}{R} = GE$
- $\Delta R = RGE = 120*2.1*0.0005 = 0.126\ \Omega$

1.Resistive Sensors

1.4 Photo conductive: semiconductors used for their property of changing resistance when electromagnetic radiation is incident on them. They are often called CdS cells (Cadmium-Sulfide) or LDR (Light Dependent Resistor)



Lux values

- Dark night (0.002)
- Living room (50)
- (32,000–100,000) Direct sun light

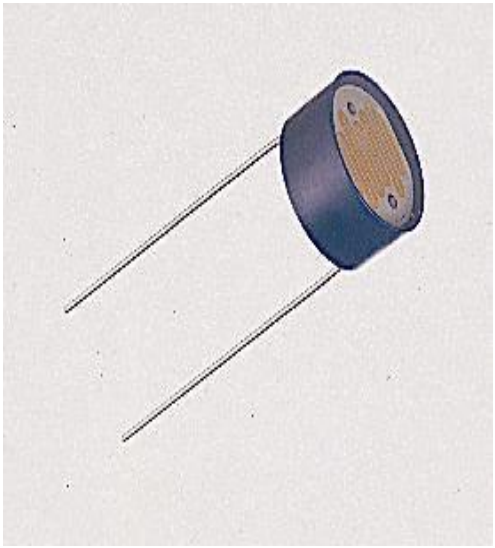


Photo transistor

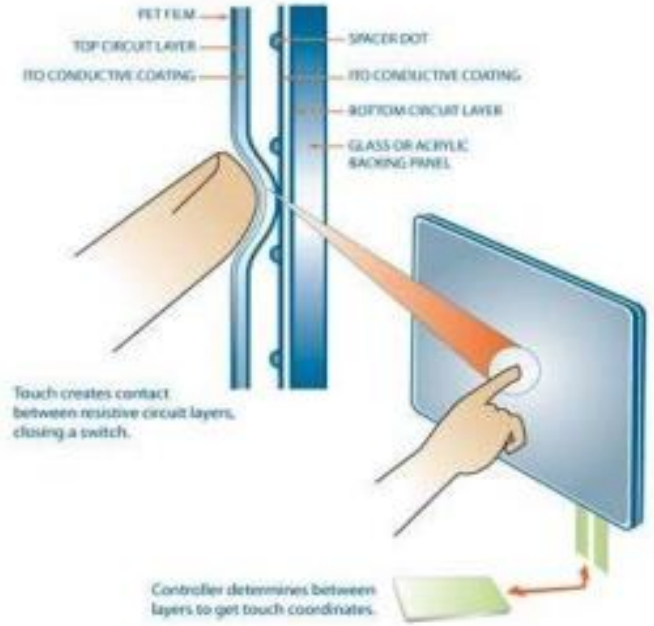
- They are p-n junctions which produce a change in current when electromagnetic radiation is incident on the junction.

- Faster and more sensitive LDR.
- Varying current rather than resistance.
- The output is either on or off

T1.Resistive Sensors

- **Resistive Touch**

The resistive touchscreen consists of a glass panel with a resistive coating plus a coversheet with a conductive coating. The two layers are separated by tiny insulating dots.
When the screen is touched, the coversheet flexes to make electrical contact with the coating on the glass. The controller alternately drives the X and Y axes on the glass layer with a +5V current and reads the resulting voltage from the cover sheet,



Wheatstone bridge

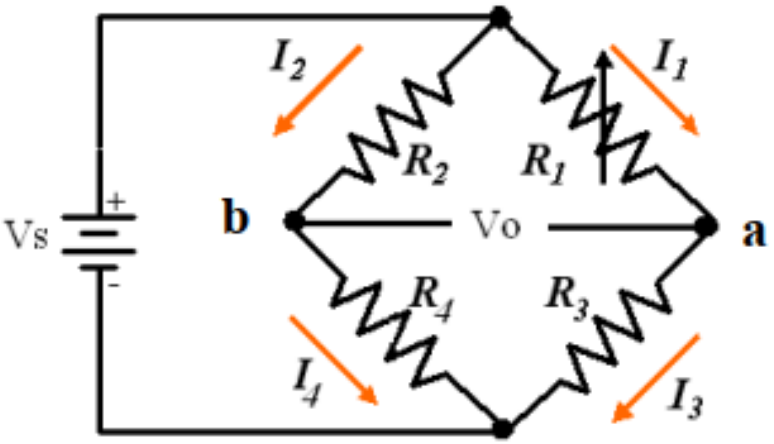
Wheatstone bridge: Consists of 4 resistors in a diamond orientation, with a resistive transducer in one or more legs.

$$I_1 = I_3 = \frac{V_S}{R_1 + R_3} \qquad I_2 = I_4 = \frac{V_S}{R_2 + R_4}$$

$$V_o = I_1 R_1 - I_2 R_2$$

$$V_o = \frac{R_1}{R_1 + R_3} V_S - \frac{R_2}{R_2 + R_4} V_S$$

if $\frac{R_1}{R_3} = \frac{R_2}{R_4}$, $V_o = 0$



$$R_1 = \frac{R_2 R_3}{R_4} \qquad \text{When } V_o = 0$$

Used to measure resistance accurately

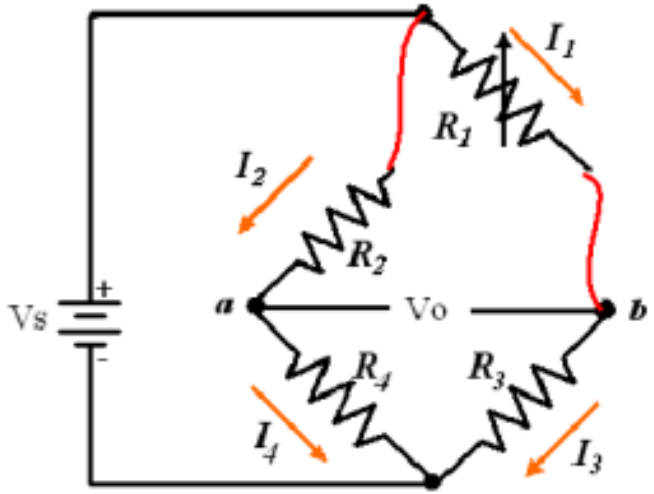
Wheatstone bridge as a sensor

If R1 is a sensor and its resistance is changed to be R1+ΔR1; therefore Vo is changed to be Vo+ΔVo

$$\Delta V_o \approx V_S \frac{\Delta R_1}{R_1 + R_3}$$

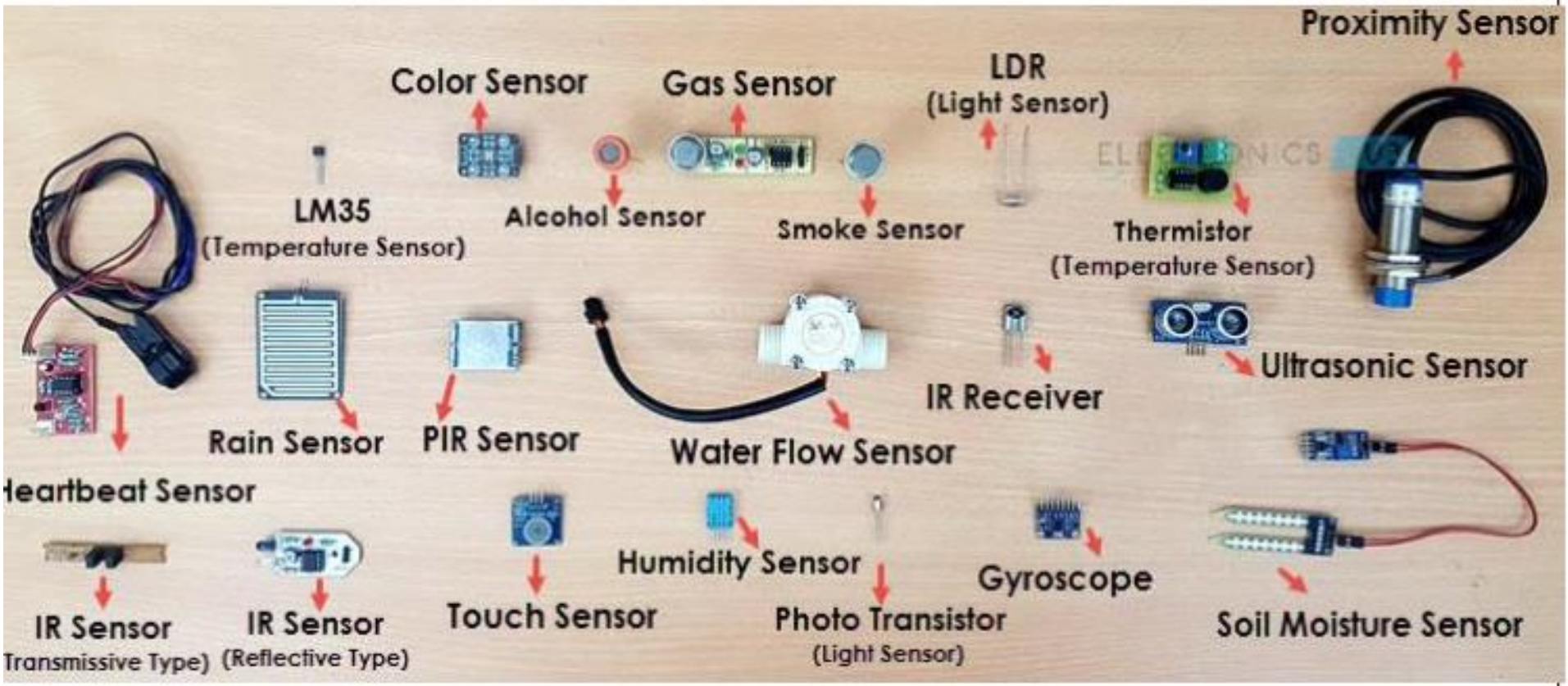
For Strain gauge $\frac{\Delta R}{R} = GE$

$$\Delta V_o = \frac{V_S R_1 GE}{R_1 + R_3}$$



Load cell

Different Types of Sensors



Real time system
Lecture 3: sensors

ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.

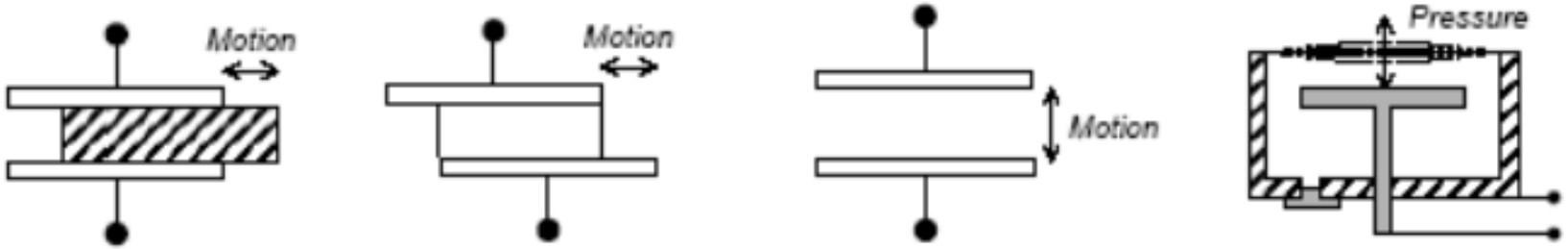
2.Capacitive Sensors

Input being measured is transformed into a capacitive change. The capacitive (C) of a parallel plate capacitor depends on the area (A), separation distance (d) and the relative permittivity (ε) of the materials between them given as:

$$C = \varepsilon_0 \frac{A}{d}$$

Vacuum permittivity

$\varepsilon_0 = 8.85 \times 10^{-12}$ F/m.

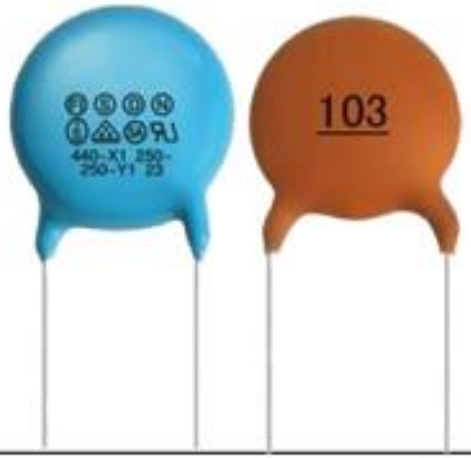


An electrolytic capacitor is made of Aluminum evaporated on either side of a very thin plastic film (or electrolyte)

- Low frequency
- High value
- Polarized

It is constructed of two or more alternating layers of ceramic and a metal layer acting as the electrodes.

- high frequency
- Lower value
- stable



- **Example:** A capacitive sensor consist of two plates in air, the plates being 50mm square and separated by a distance of 1mm. A new sheet of dielectric material of thickness 1mm and 50mm square can slide between the plates. Determine the capacitance of the sensor when the sheet has been displaced so that half of it is between the capacitor plates. The dielectric of the new sheet is 4 and it can be presumed as 1 for the air.
- $C_{Total} = C_{air} + C_{new}$

$$C_{air} = \frac{E_o E_r A}{D} = 8.85 * 10^{-12} * 1 * 50 * 25 * 10^{-3^2} / 1 * 10^{-3}$$

$$= 1.106 * 10^{-11} = 11.06 \text{ pf}$$

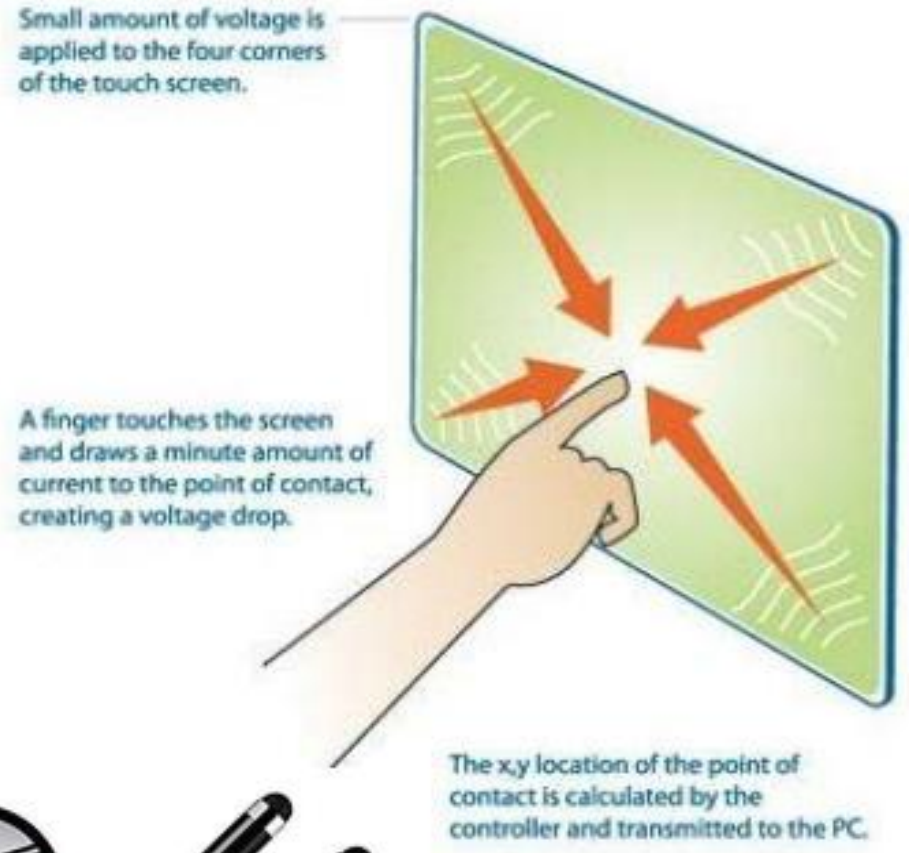
$$C_{new} = \frac{E_o E_r A}{D} = 8.85 * 10^{-12} * 4 * 50 * 25 * 10^{-3^2} / 1 * 10^{-3}$$

$$= 4.425 * 10^{-11} = 44.25 \text{ pf}$$

$$C_{Total} = 11.06 + 44.25 = 55.31 \text{ pf}$$

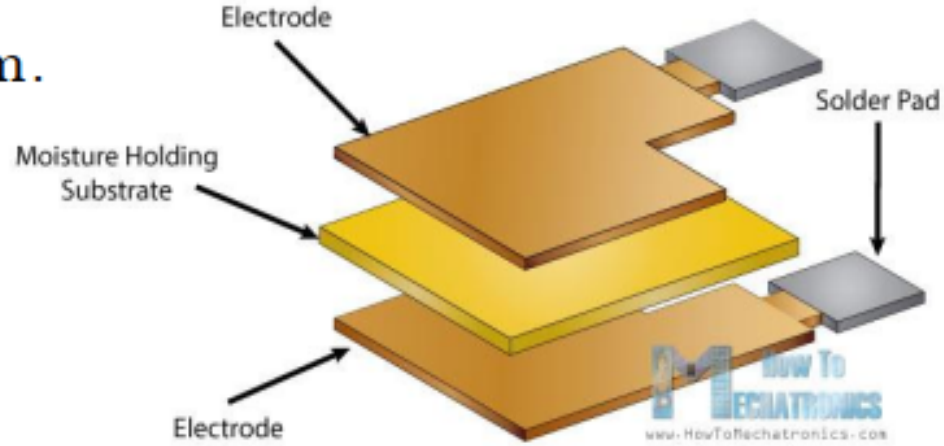
Capacitive Touch Screen

- Capacitive touchscreens work by sensing the conductive properties of an object, usually the skin.
- When a capacitive panel is touched, a small amount of charge is drawn to the point of contact.
- A controller measures the current from different corners to determine the location



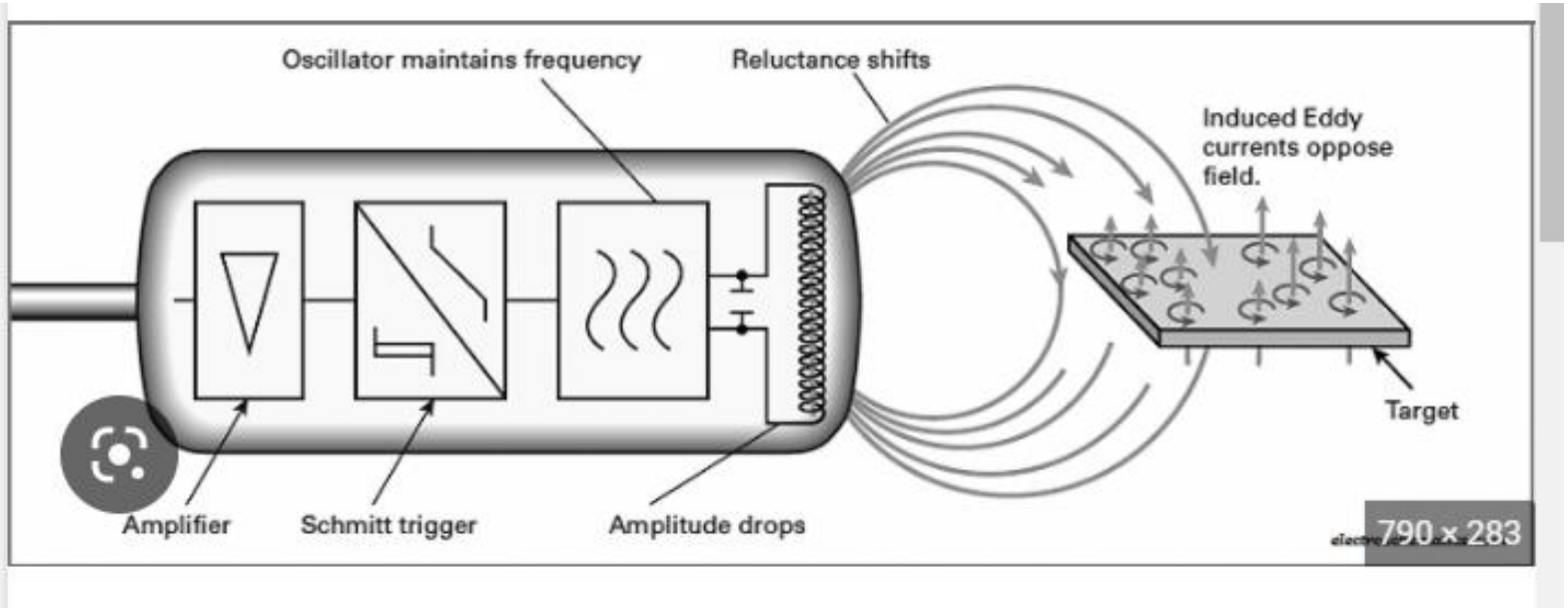
Humidity Sensor

- **Capacitive**: A capacitive humidity sensor measures relative humidity by placing a thin strip of metal oxide between two electrodes. The metal oxide's electrical capacity changes with the atmosphere's relative humidity. Weather, commercial and industries are the major application areas
- **Resistive**: Resistive humidity sensors utilize ions in salts to measure the electrical impedance of atoms. As humidity changes, so does the resistance of the electrodes on either side of the salt medium.



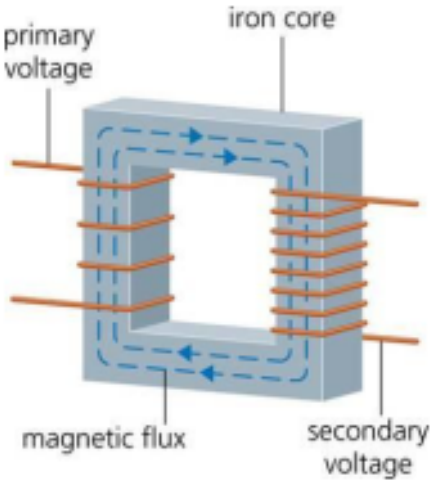
3- Inductive sensor

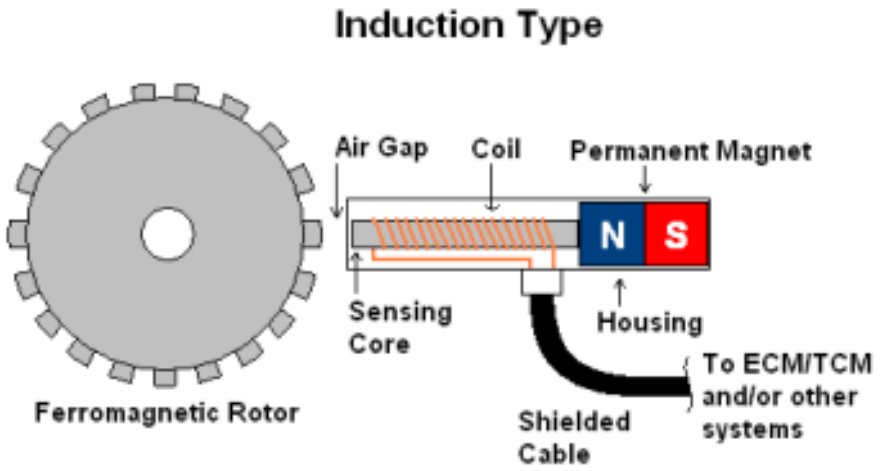
- An inductive sensor is a device that uses the principle of electromagnetic induction to detect or measure objects.
- An inductor develops a magnetic field when a current flows through it; alternatively, a current will flow through a circuit containing an inductor when the magnetic field through it changes.
- This effect can be used to detect metallic objects that interact with a magnetic field.


Oscillator maintains frequency
Reluctance shifts
Induced Eddy currents oppose field.
Target
Amplifier
Schmitt trigger
Amplitude drops
790 × 283

3. Inductive Sensors Applications

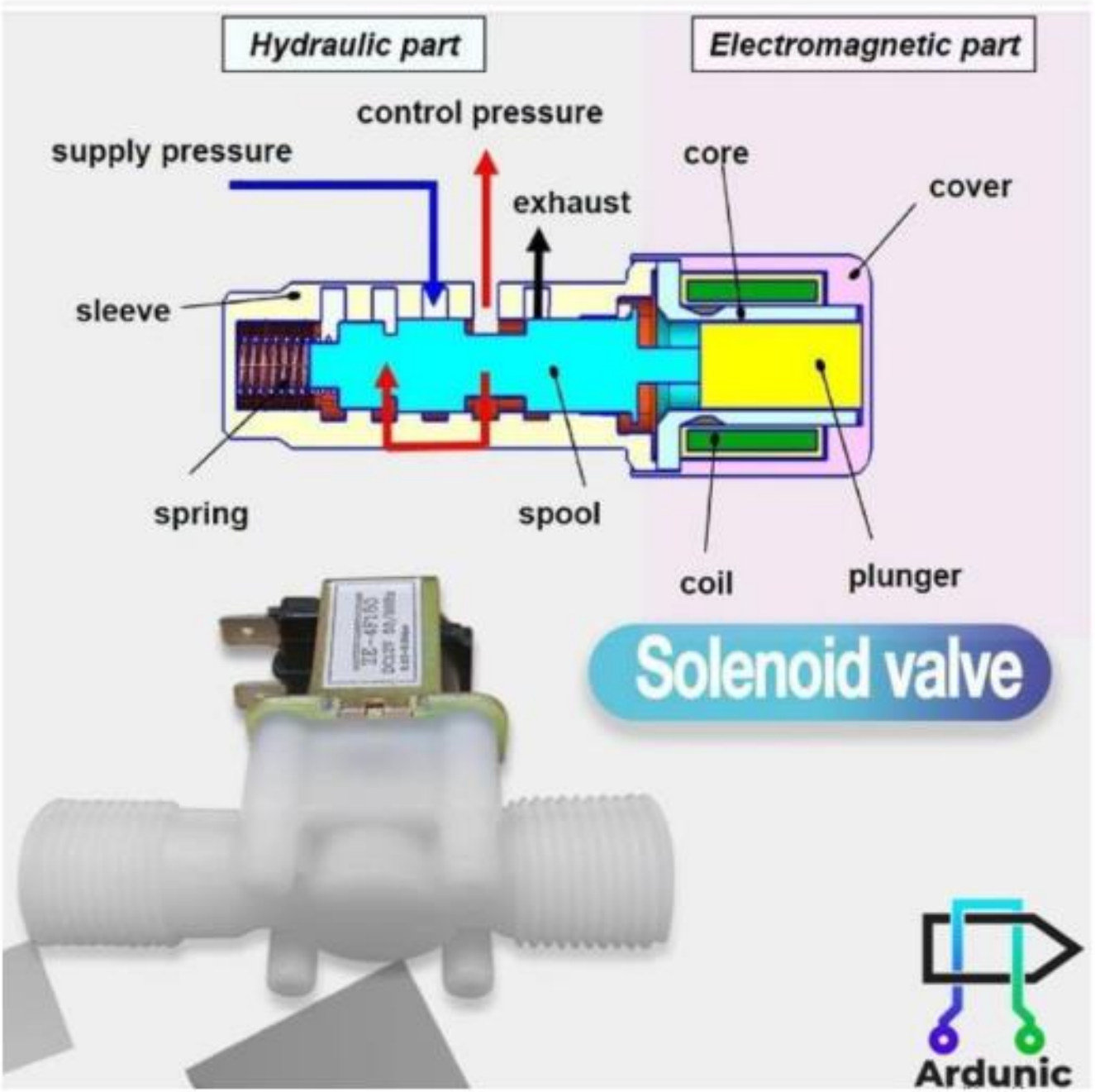
- Detection of ferrous metals - steel, iron, cobalt, nickel
- Determine the position of a mechanical moving object
- Speed calculator
- Coil and transformer production





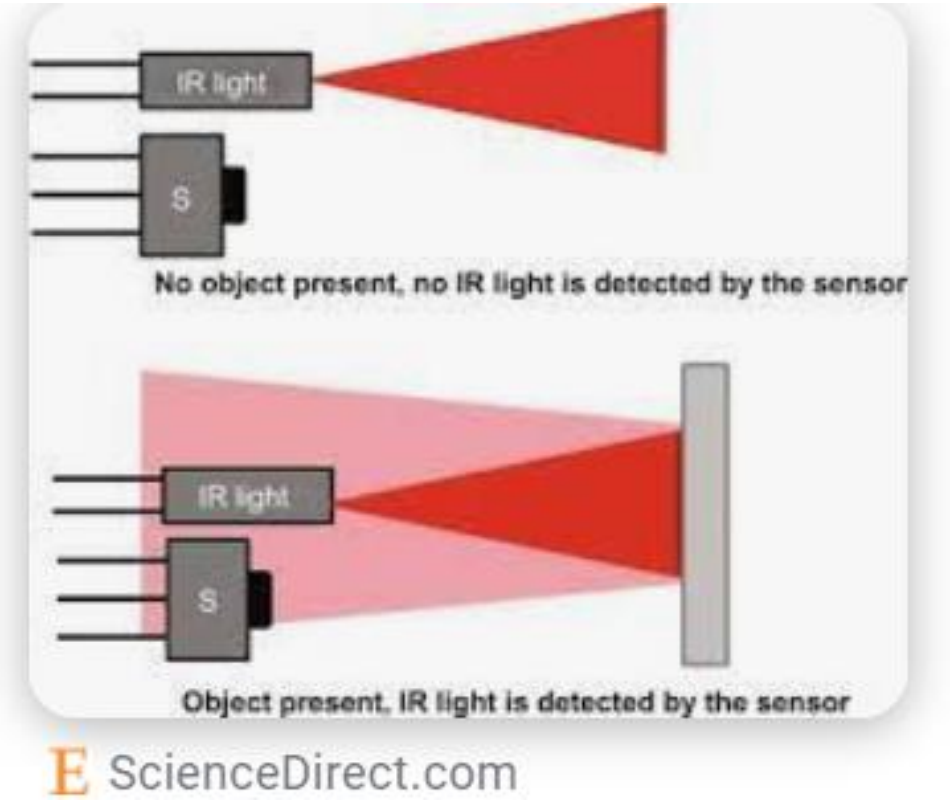
• Solenoid valve

- A solenoid valve is an electrically operated valve that is mainly used to regulate the flow of air or a liquid in pneumatic and hydraulic power systems. Different types of solenoid valves are available in the market based on application.


Hydraulic part
Electromagnetic part
control pressure
supply pressure
exhaust
core
cover
sleeve
spring
spool
coil
plunger
Solenoid valve
Ardunic

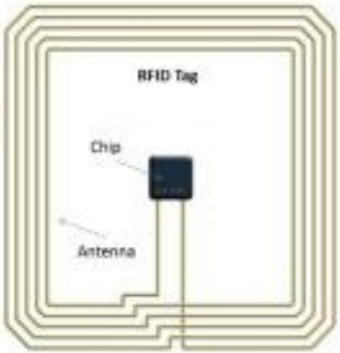
- # Optical sensor

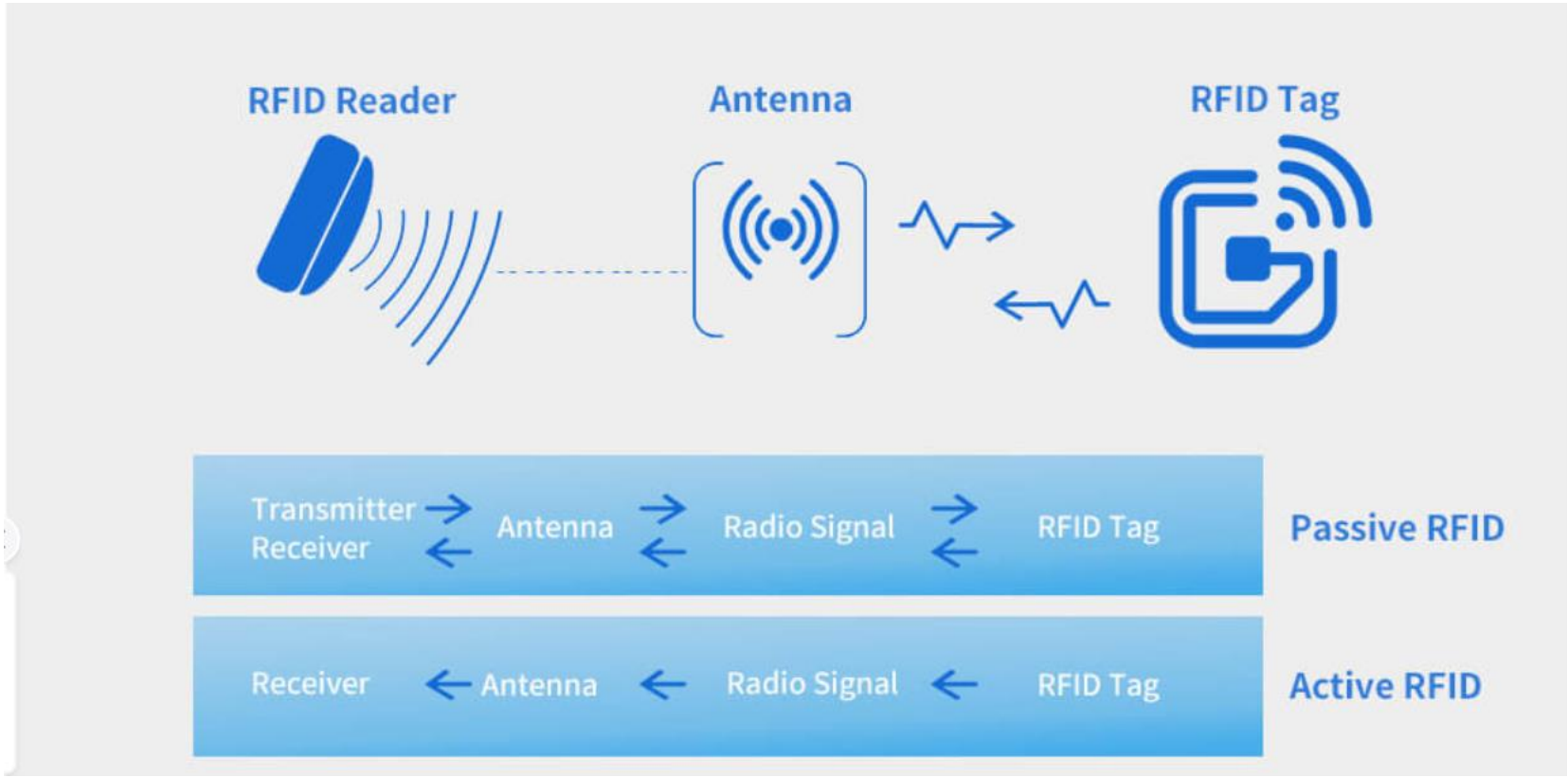
An optical sensor converts light rays into electronic signals. It measures the physical quantity of light and then translates it into a form that is readable by an instrument



Radio-frequency identification (RFID)

- uses electromagnetic field to automatically identify and track tags attached to objects.
- Passive tags collect energy from a nearby RFID reader.
- Active tags have a local power source (such as a battery) and may operate hundreds of meters from the RFID reader.



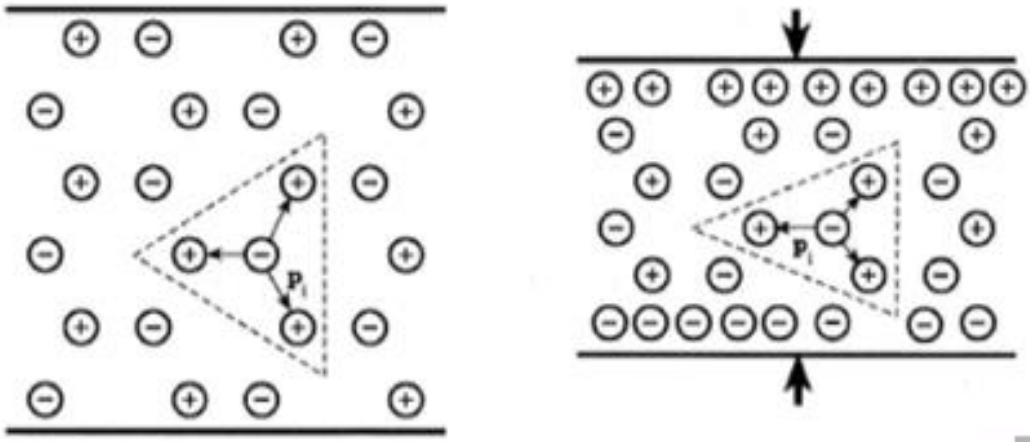


APPLICATION

- Inventory control
- Equipment tracking
- Personnel tracking
- Ensuring that patients receive the correct medications and medical devices
- Monitoring patients
- Providing data for electronic medical records systems

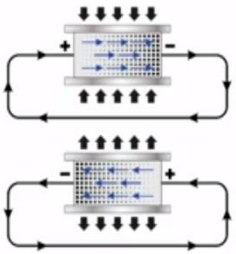
4.Piezoelectric Sensors

- **Piezoelectricity:** some dielectric materials when stretched its surfaces become charged

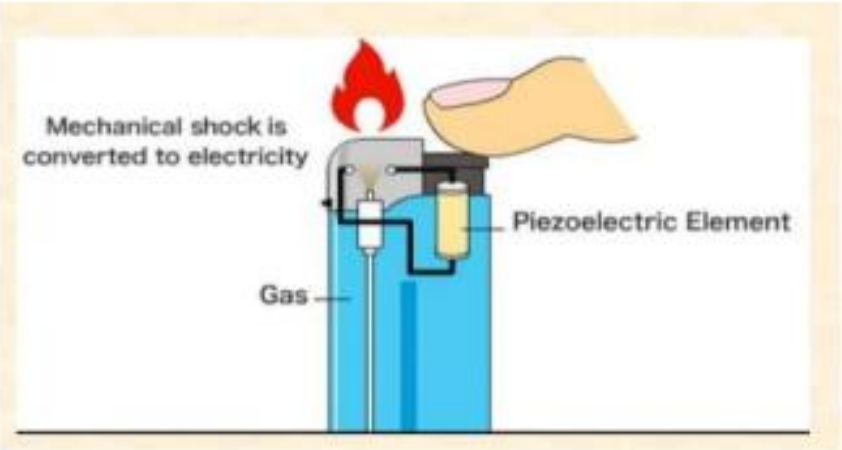


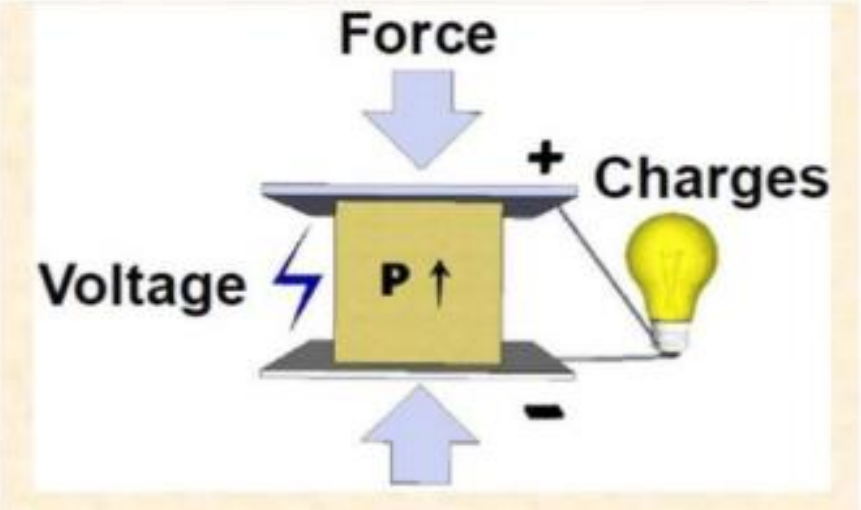
Strain causes a redistribution of charges and results in a net electric

- A piezoelectric material produces voltage by distributing charge (under mechanical strain/stress)



Energy harvesting





The other way round!

- If you pass electricity through the same crystals, they "squeeze themselves" by vibrating back and forth!
- Can be used in ultrasound equipment, a piezoelectric transducer converts electrical energy into extremely rapid mechanical vibrations.
- In quartz clock

• ultrasonic sensor

- an ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear)

- 85 to 180 Hz (male)
- 165 to 255 Hz (female)

Ultra-sonic sensor

Ultrasonic Sensor

speed of sound:

$v = 340\ m/s$

$v = 0{,}034\ cm/\mu s$

$Time = distance/speed$

$$distance = \frac{time}{2} * speed$$

Fire Alarm

- An **optical smoke detector** contains a source of infrared, visible, or ultraviolet light, a lens, and a photoelectric receiver.
- All of these components are arranged inside a chamber.
- Piezo-electric loud speaker is used.

Carbon monoxide detector

- Carbon monoxide is an odorless, colorless, and tasteless gas that is near impossible to identify without a proper detector.

- It is caused by fuels not burning completely

- Metal-oxide detectors have open chambers containing sensors made of metal (tin or platinum) oxide.

- Usually the percent is 0.2 ppm (particle per million) for clear air
- It shouldn't exceed 35 ppm for more than one hour

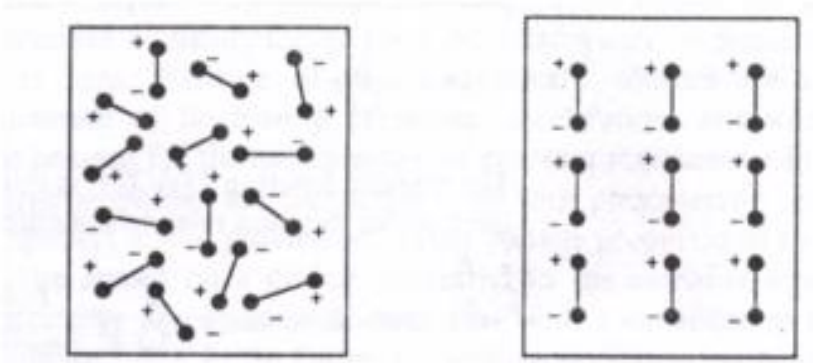
Carbon monoxide detector

- When there's carbon monoxide around, the metal oxide reacts with it: the carbon monoxide "steals" oxygen from the metal oxide, converting itself into carbon dioxide, turning the metal oxide into pure metal, and producing heat at the same time.

- An electronic circuit monitors the temperature inside the chamber and sounds the alarm if too much heat is produced too quickly.

6.Pyroelectric Sensors

- Pyroelectricity can be described as the ability of certain materials to generate a temporary voltage when they are heated or cooled.
- When the pyroelectric material is exposed to infrared radiation its temperature rises and the amount of polarization is reduced.
- Charge decreases as temperature increases.

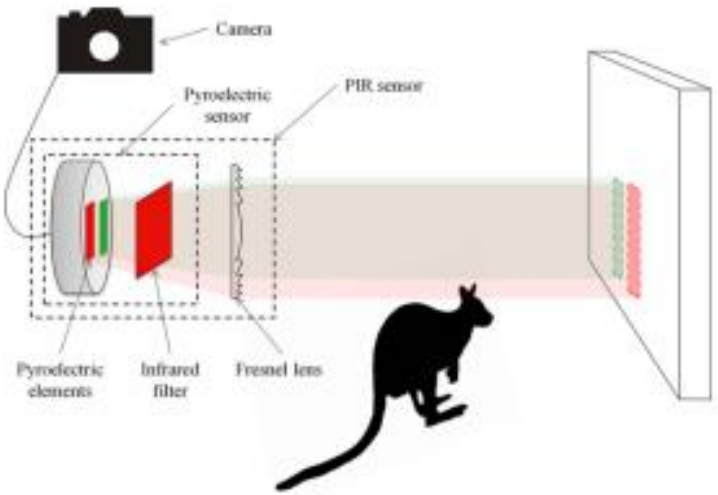


(a) Before polarisation (b) after polarisation

- Used to measure sensitive temperature changes.
- Stable and can tolerate severe conditions (flame detector)
- Near-infrared detector (trap camera)

Pyroelectric sensor's applications:

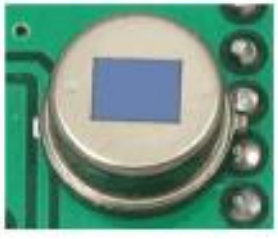
- Security
- Light control system
- Motion activated camera
- Pyrometer



Passive InfraRed Sensor (PIR)

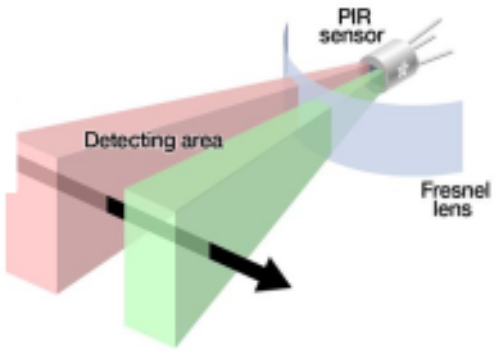
- All objects with a temperature above Absolute Zero (0 Kelvin -273.15 °C) emit heat energy in the form of infrared radiation, including human bodies. The hotter an object is, the more radiation it emits.

- Passive infrared (PIR) sensors are sensitive to a person's skin temperature through emitted body radiation at mid-infrared wavelengths

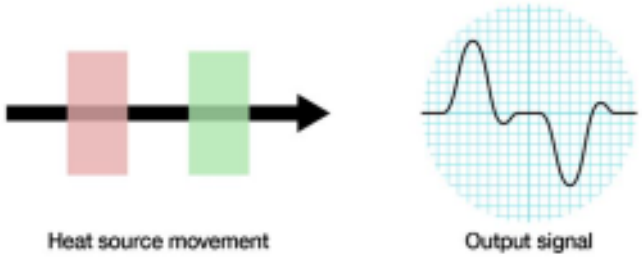


Passive InfraRed Sensor (PIR)

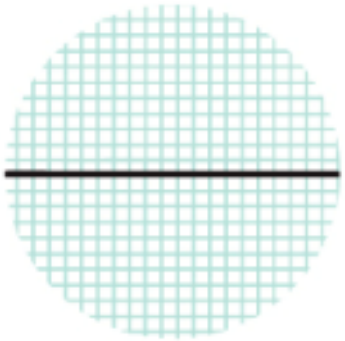
- When the sensor is idle, i.e. there is no movement around the sensor; both slots detect the same amount of infrared radiation, resulting in a zero output signal.

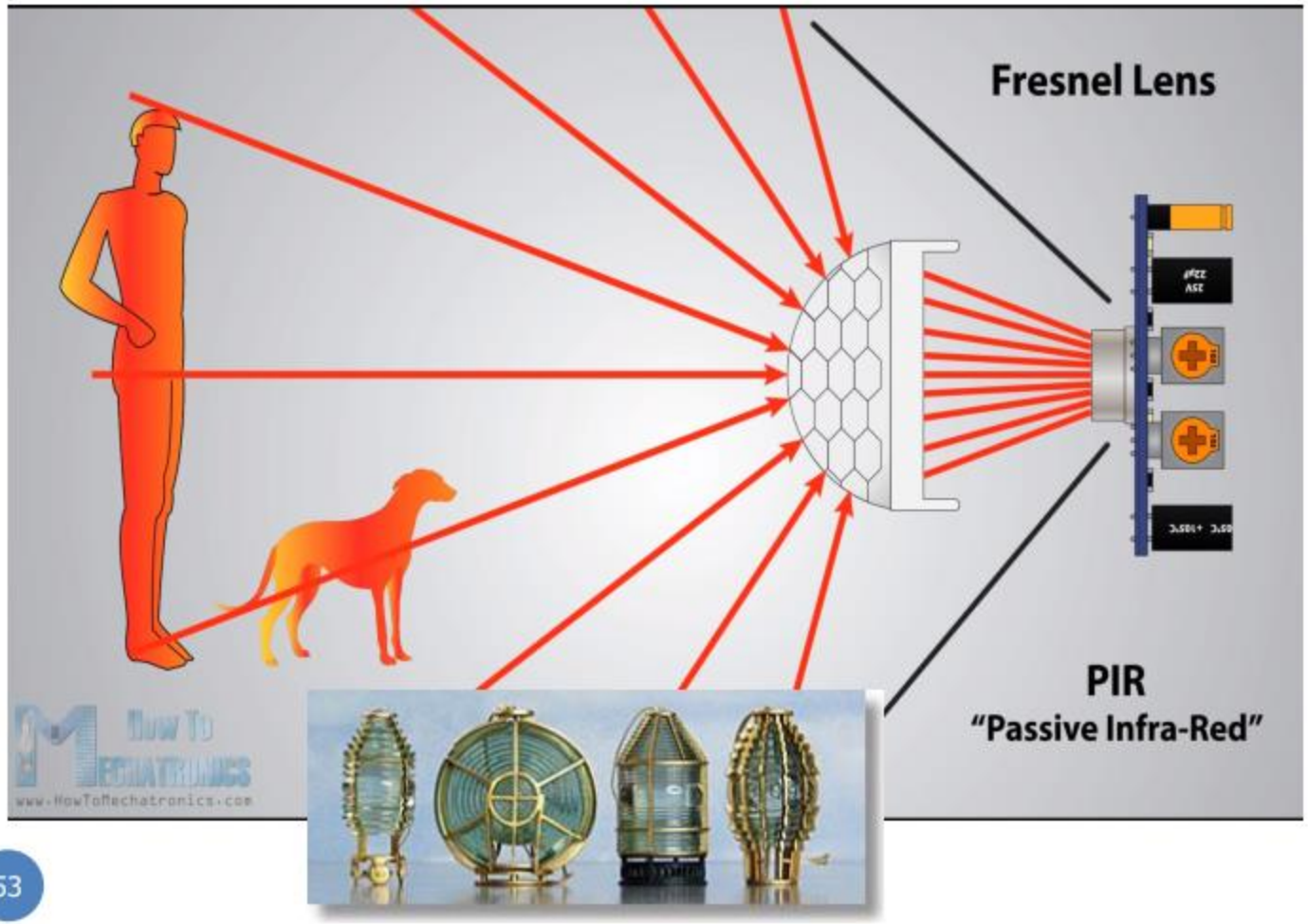


- But when a warm body like a human or animal passes by; it first intercepts one half of the PIR sensor, which causes a positive differential change betwee two halves.



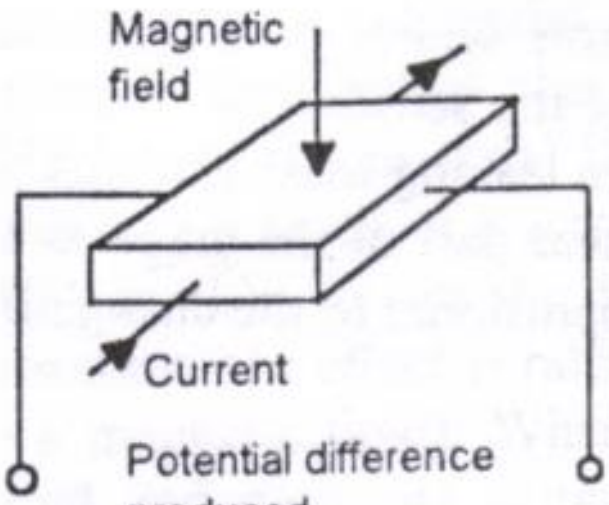
No energy is emitted from the sensor, thus the name **passive infrared**

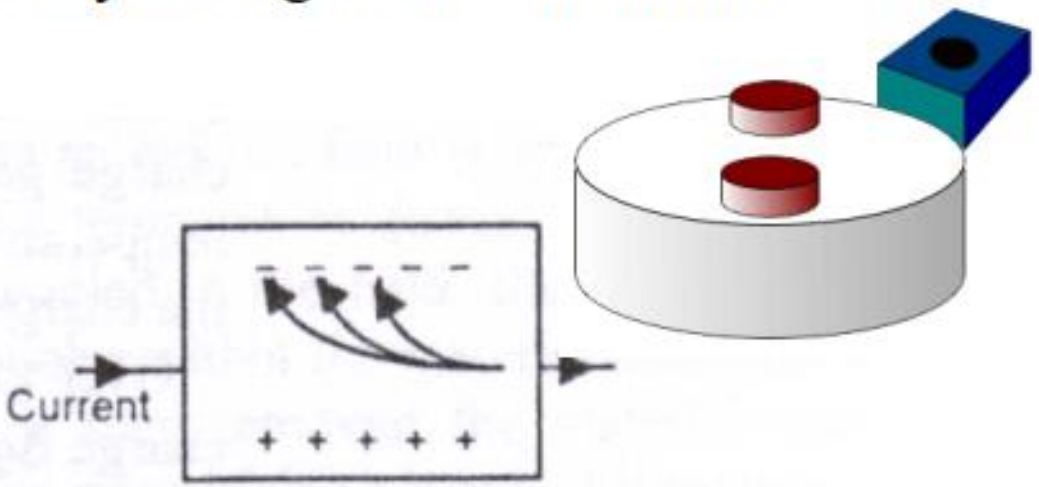



Fresnel Lens
PIR
"Passive Infra-Red"
How To Mechatronics
www.HowToMechatronics.com

7.Hall effect Sensors

The action of a magnetic field on a flat plate carrying an electric current generates a potential difference which is a measure of the strength of the field. A beam of charged particles can be deflected by a magnetic field (Hall effect).





Note that electrons move in opposite direction to arrow indicating current direction

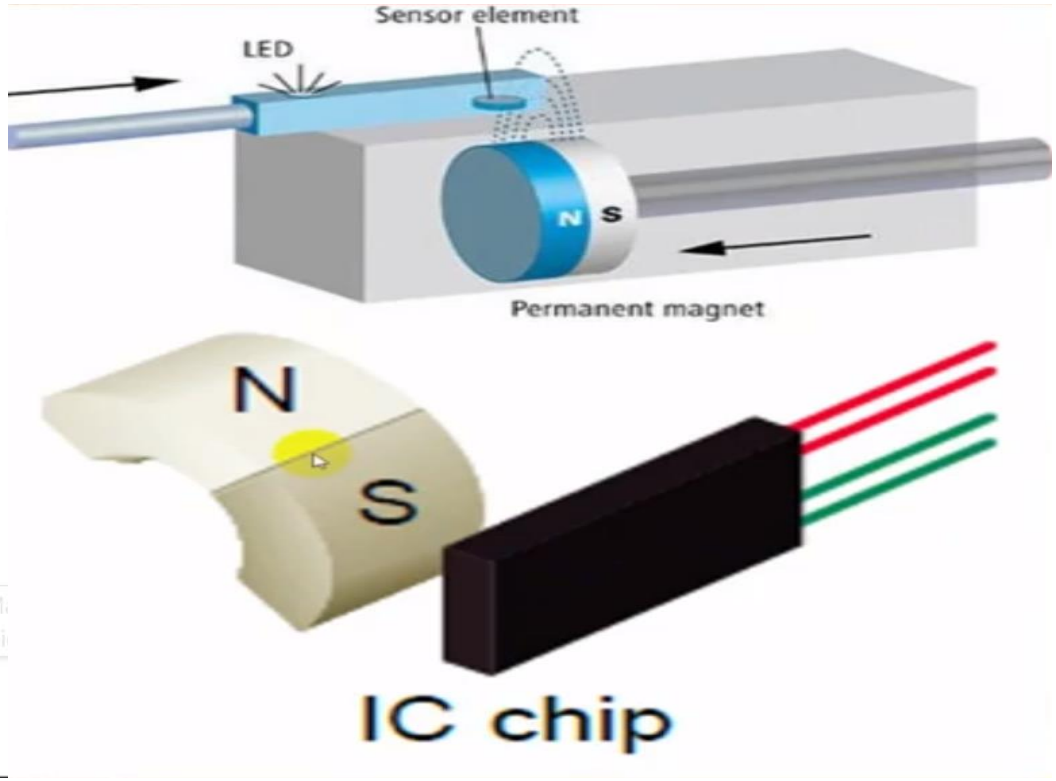
• Hall effect sensors

A Hall effect sensor (or simply Hall sensor) is a type of sensor which detects the presence and magnitude of a magnetic field using the Hall effect. The output voltage of a Hall sensor is directly proportional to the strength of the field. It is named for the American physicist Edwin Hall

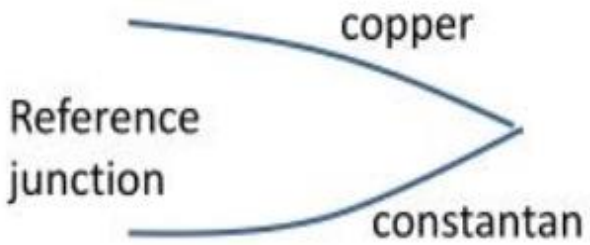
Applications

- Electronics compass
- Position sensing of moving objects

Hall effect Magnetometer for Arduino

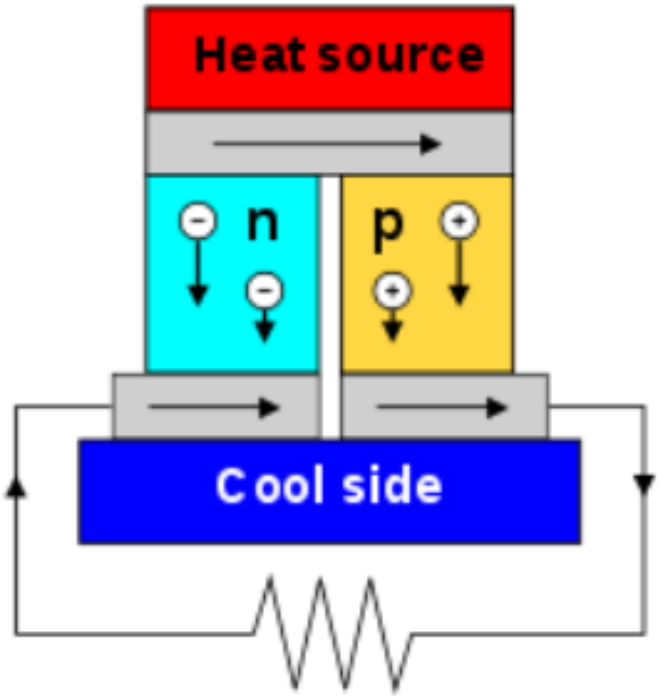


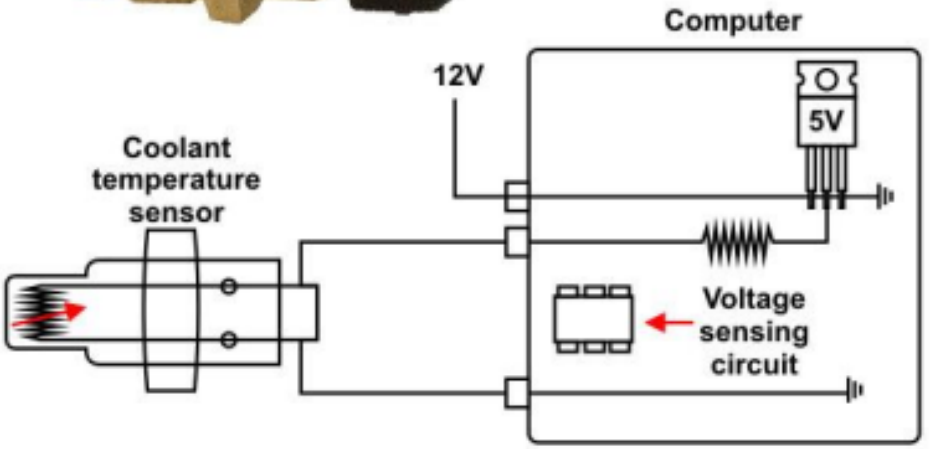
Thermoelectric sensors



- Thermocouple
  - Physical basis-
    - 2 different metals joined together to make a circuit
    - Electrons flow from one metal to another until a voltage difference V typical for those metals and environmental temperature is reached
  - Temperature difference between junctions provides a relative measure of the voltage difference
  - One junction must have a known reference temperature
  - Range of copper-constantan Type T: -75 to 200C
  - Voltage difference is small: ~40μV/C for Type T
- Thermopile- thermocouples in series to amplify voltage difference

Principle and applications



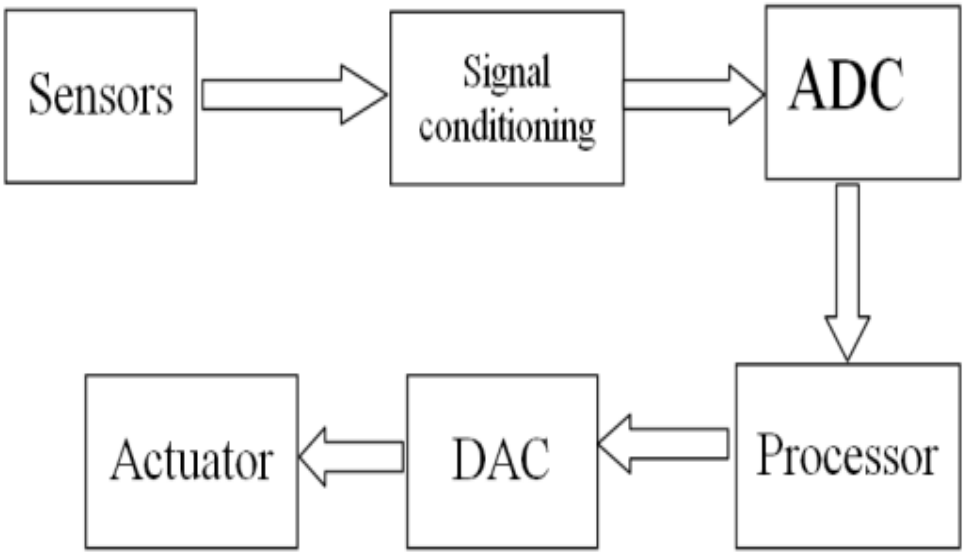


- Heat sensor (metal)
- Power generator (semiconductor)
  - smartwatch powered by body heat

Real time system
Lecture 4: signal conditioning circuit (SCC)

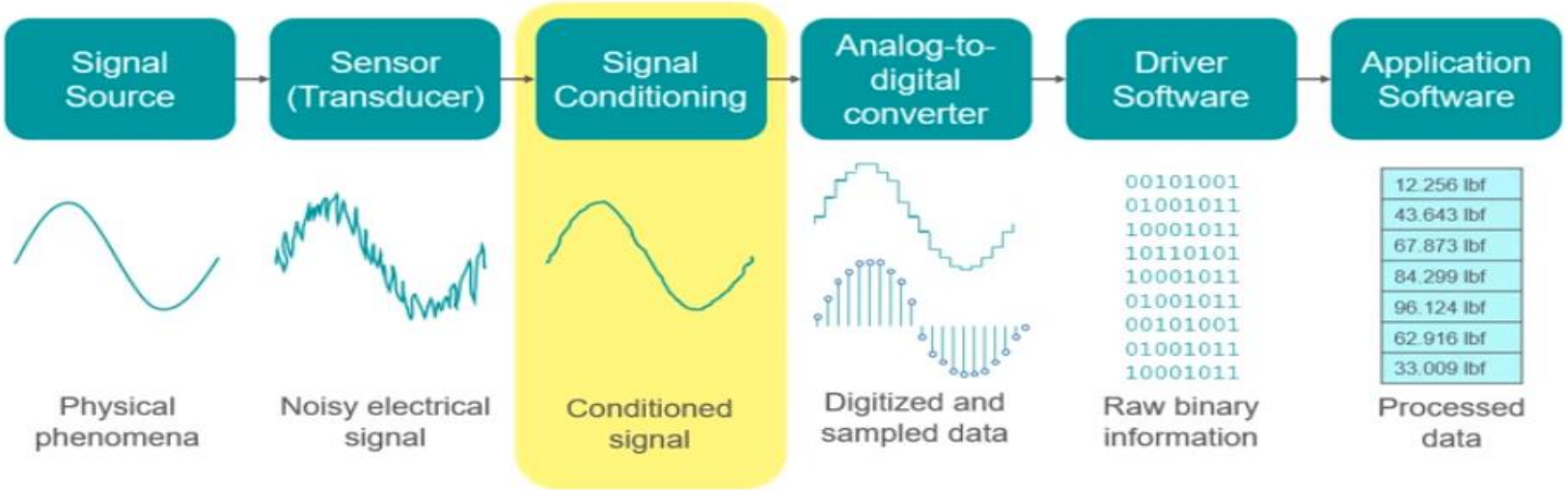
ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.



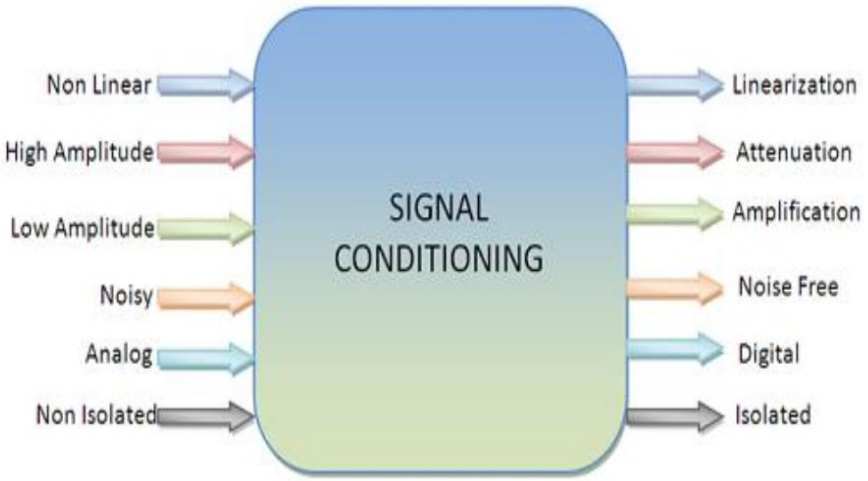
System

What is Signal Conditioning?



Signal Conditioning

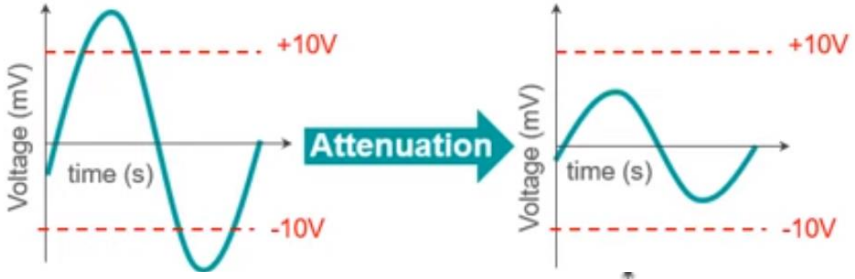
- Signal conditioning refers to operations performed on signals to convert them to a form suitable for interfacing with other elements in the process control loop.
- Signal conditioning circuits are used to process the output signal from sensors of a measurement system to be suitable for the next stage of operation



Types of Signal Conditioning

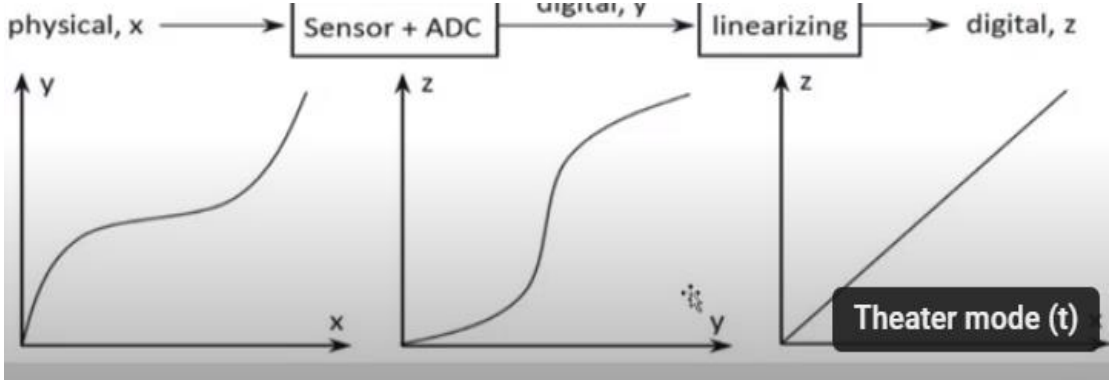
Amplification
Ex. Thermocouples, strain gauges

Attenuation
For voltages >10V



Filtering
To block out noise, aliasing

Linearization

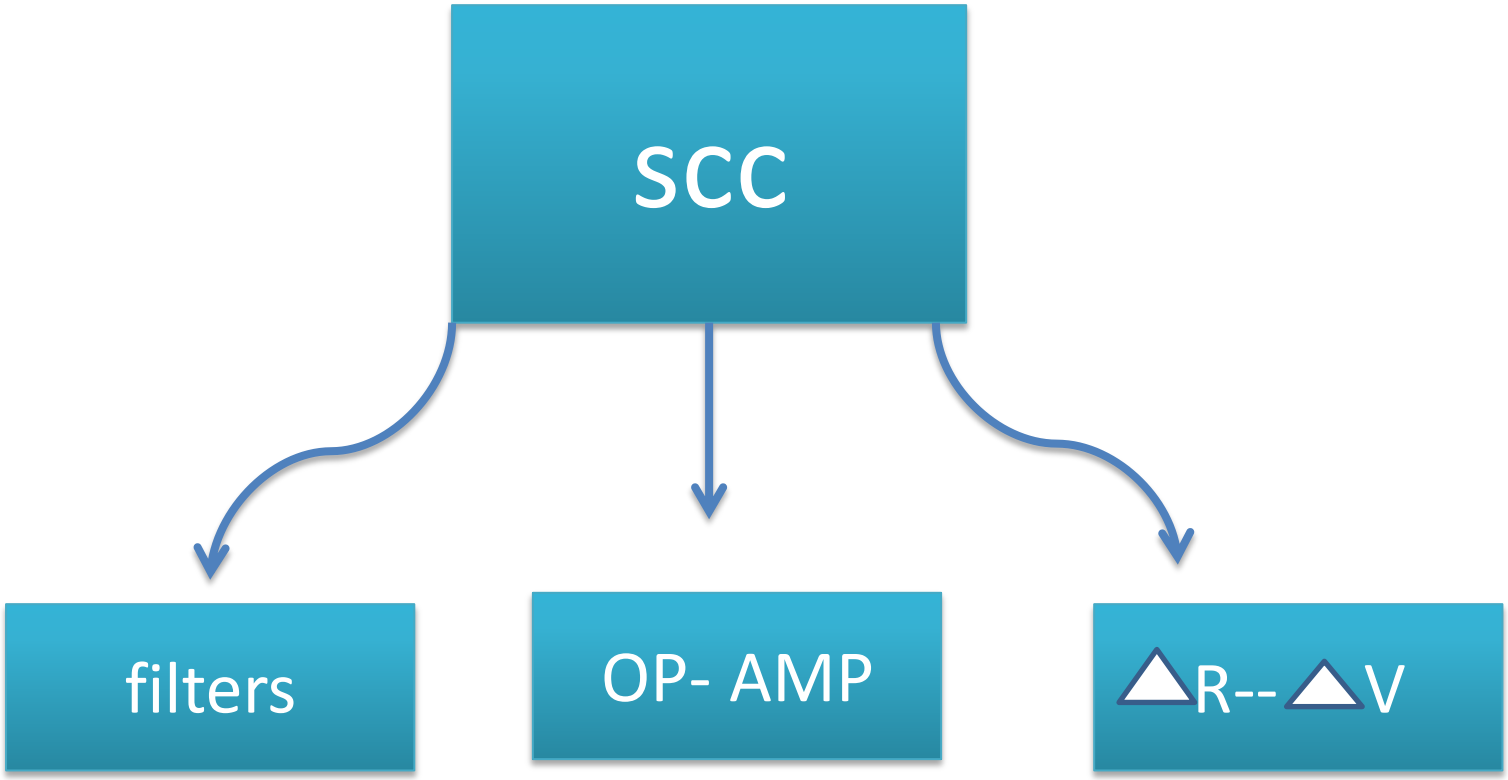


Signal Conditioning

- Some functions of signal conditioners are:

1. Adjusting of the sensor signals
2. Conversion of currents to voltages
3. Supply of (ac or dc) excitations to the sensors so changes in resistance, inductance, or capacitance are converted to changes in voltage
4. Filtering to eliminate noise or other unwanted signal components
5. Making a nonlinear signal a linear one

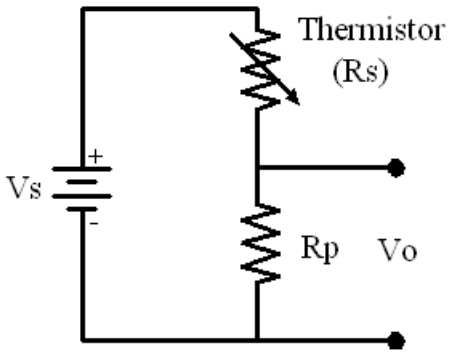
some of Signal Conditioning circuits



Resistance to Voltage Conversion

1.Potentiometer: used to transfer changes in resistance (Thermistor or strain gauge) to voltage

Vo=Vs*Rp/(Rp+Rs)



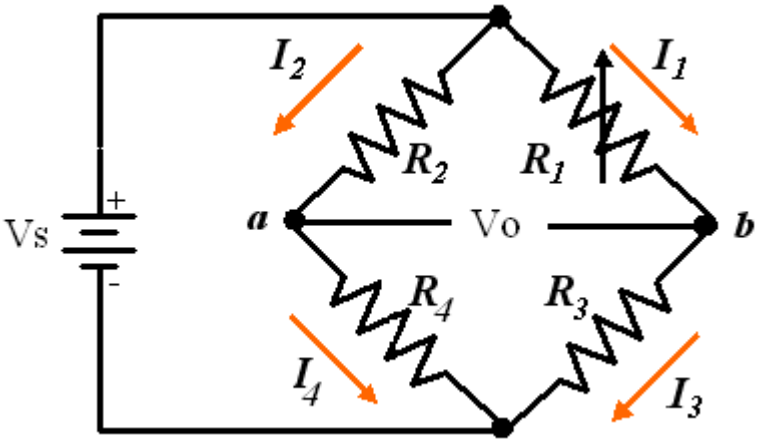
2.Wheatstone bridge: Consists of 4 resistors in a diamond orientation, with a resistive transducer in one or more legs.

$$I_1 = I_3 = \frac{V_S}{R_1 + R_3} \qquad I_2 = I_4 = \frac{V_S}{R_2 + R_4}$$

$$V_o = I_1 R_1 - I_2 R_2$$

$$V_o = \frac{R_1}{R_1 + R_3} V_S - \frac{R_2}{R_2 + R_4} V_S$$

if $\frac{R_1}{R_3} = \frac{R_2}{R_4}$, $V_o = 0$



Resistance to Voltage Conversion

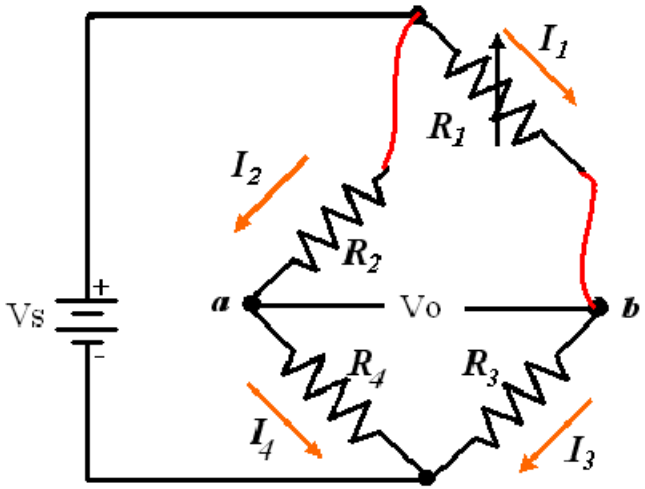
If R1 is a sensor its resistance is changed to be $\Delta R1$ therefore Vo is changed to ΔVo

$$V_o = V_S\left[\frac{R_1}{R_1+R_3}-\frac{R_2}{R_2+R_4}\right]$$

$$\Delta V_o = V_S\left[\frac{R1R2+R1R4-R1R2-R2R3}{(R1+R3)(R2+R4)}\right]$$

$$\Delta V_o \approx V_S\left[\frac{R1R4-R2R3}{(R1+R3)(R2+R4)}\right]$$

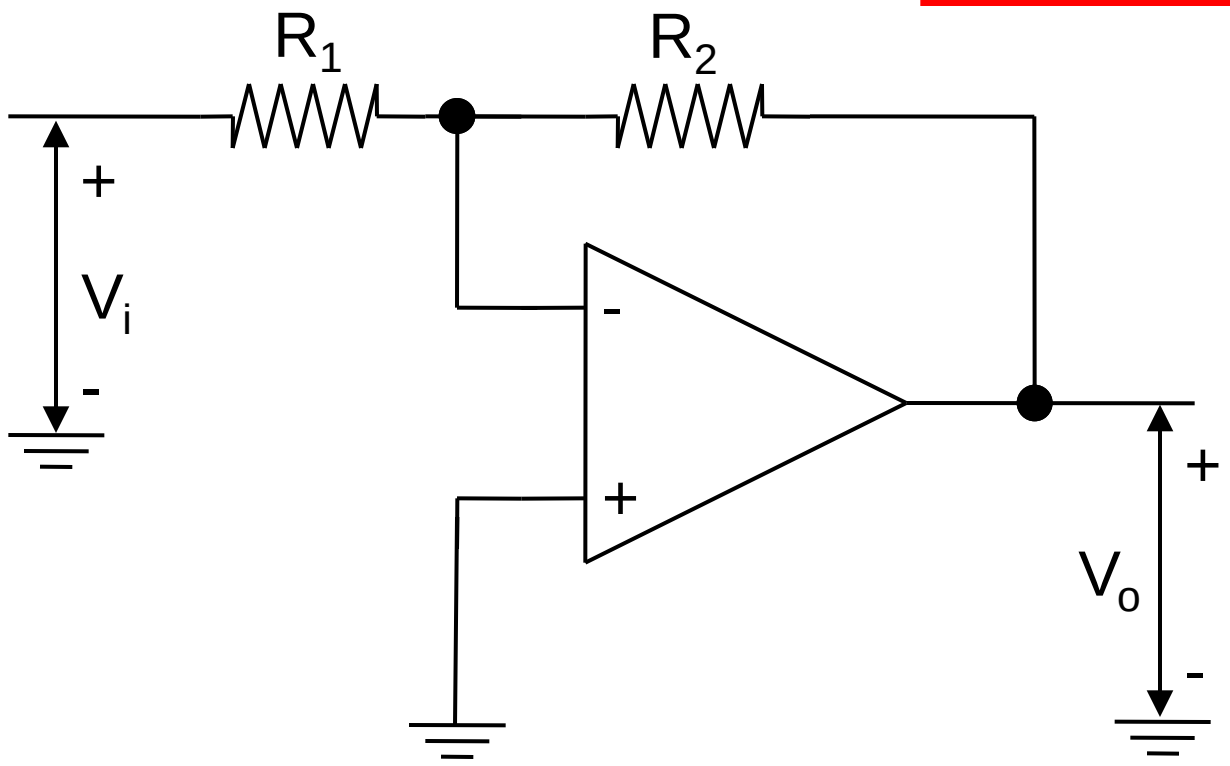
The resistance of wires connecting R1 will be effected by temp. so the connection shown is used so the effect of temp. is added to the two opposite branches



Operational Amplifier for Signal Conditioning

1. Inverting Amplifier:

$$V_0 = -\frac{R_2}{R_1} V_i$$

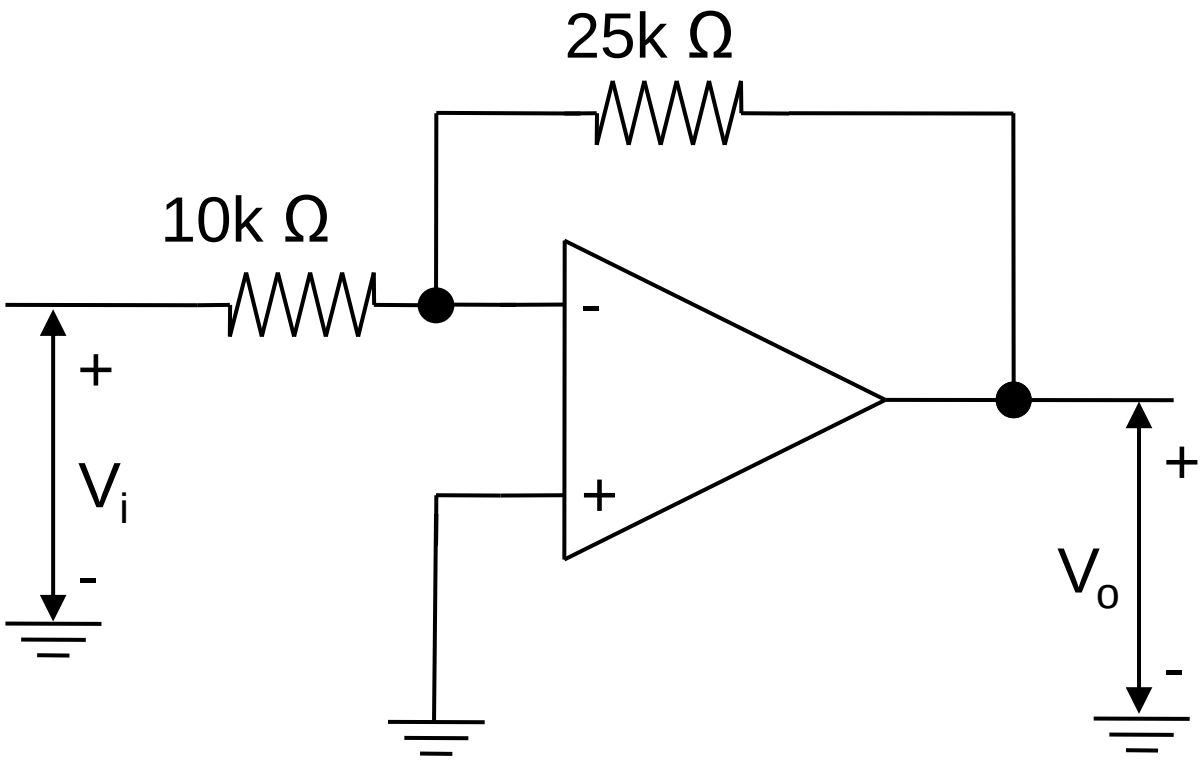


Ex : 1

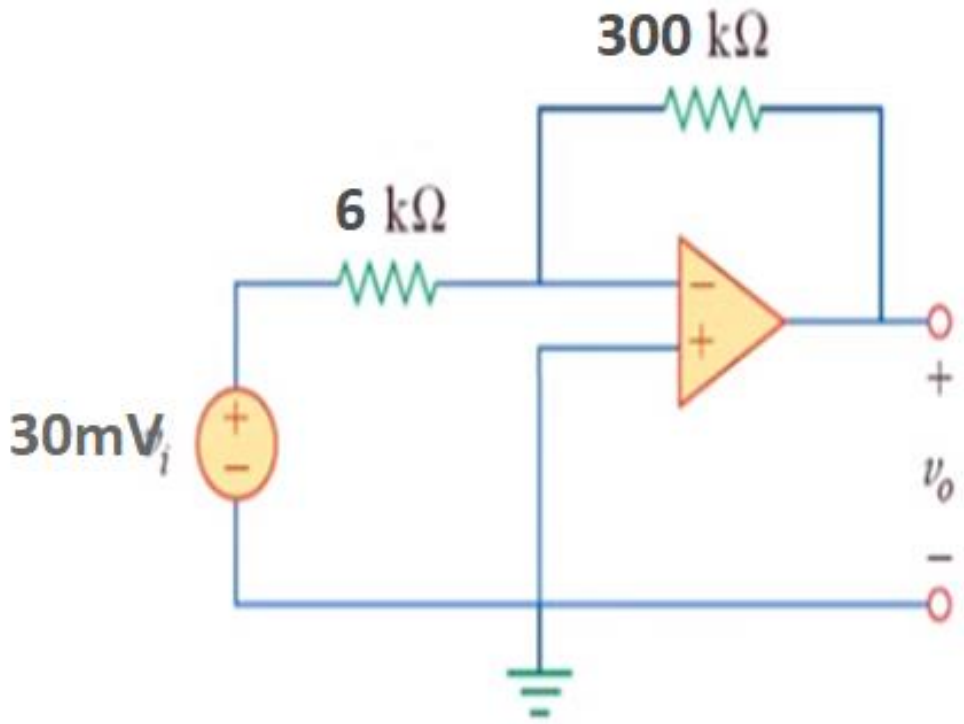
The op amp in figure bellow if Vi= 0.5 v calculate :

a) The output voltage V0 .

b) The current in the 10kΩ resistor.



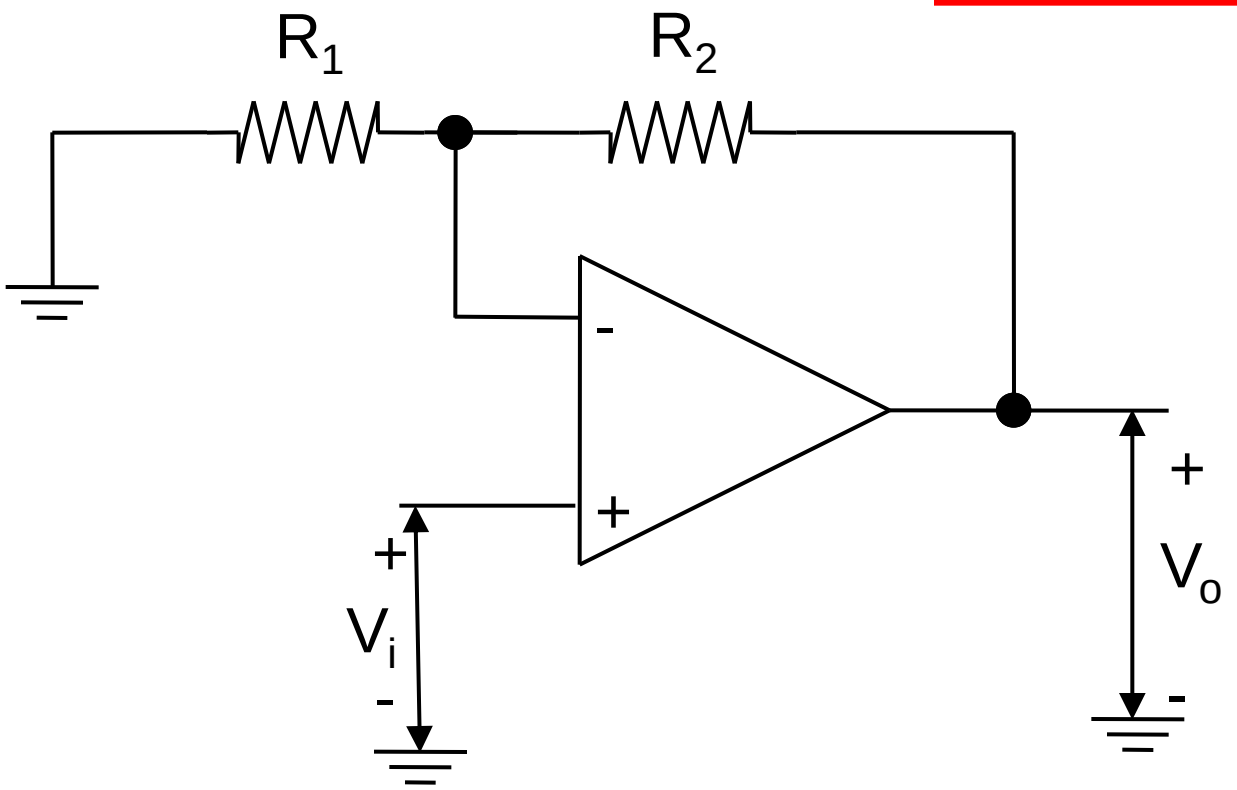
Ex: 2 Find the Vo



Operational Amplifier for Signal Conditioning

2. Non-Inverting Amplifier:

$$V_o = \left(1 + \frac{R_2}{R_1}\right) V_i$$



Operational Amplifier for Signal Conditioning

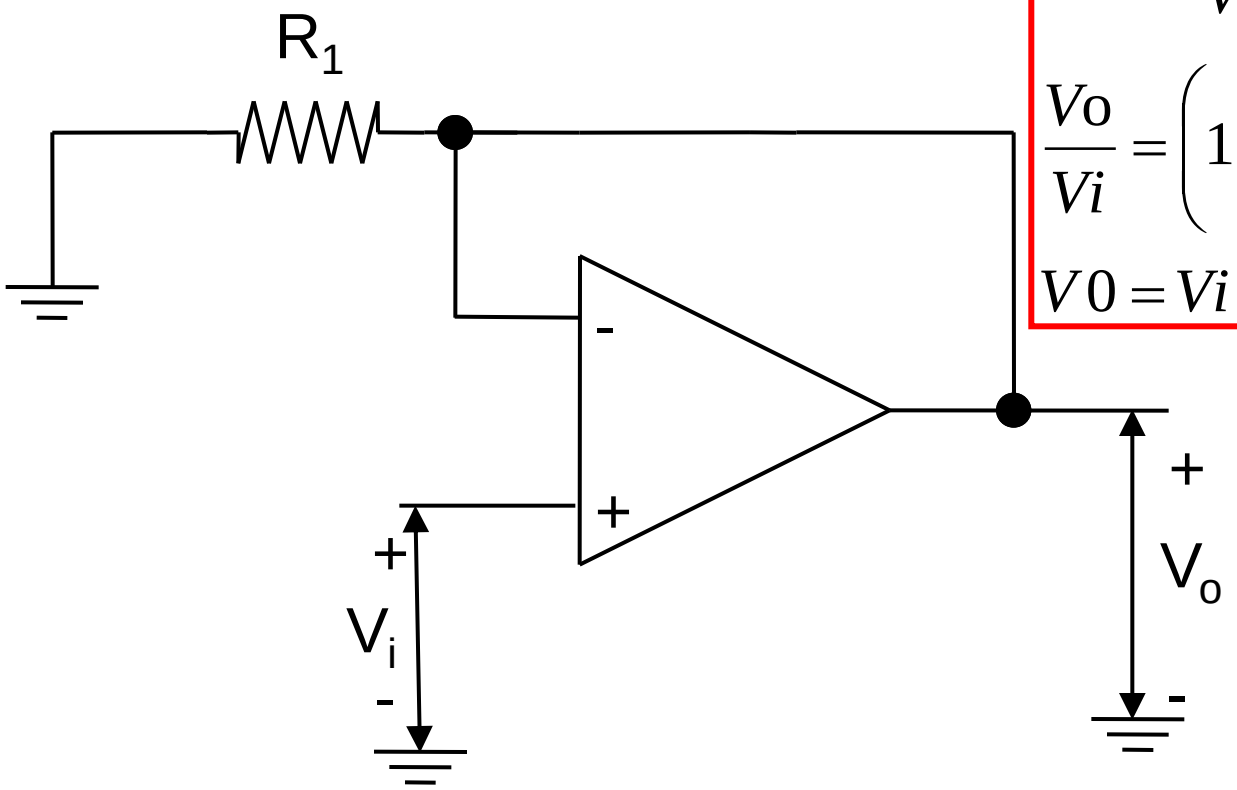
3. Voltage follower –buffer

$$V_o = \left(1 + \frac{R_2}{R_1}\right) V_i$$

$$AV = \frac{Vo}{Vi} = \left(1 + \frac{R_2}{R_1}\right)$$

$$\frac{Vo}{Vi} = \left(1 + \frac{0}{R_1}\right)$$

$$V0 = Vi$$

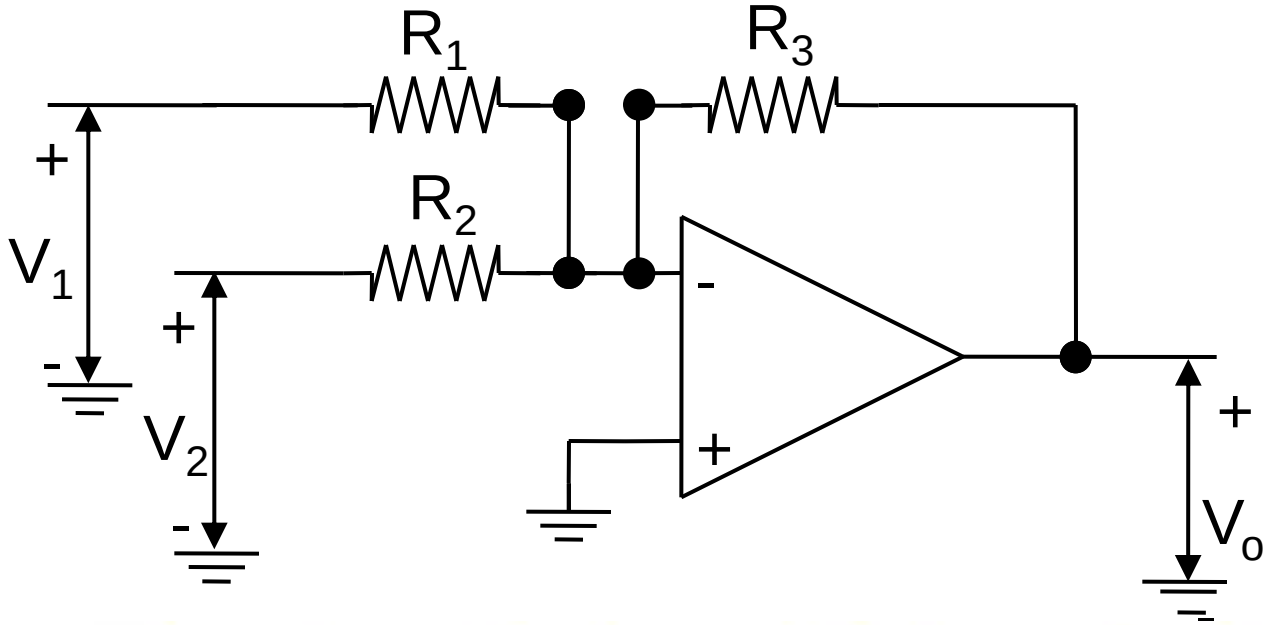


Operational Amplifier for Signal Conditioning

4. Summing Amplifier

Besides amplification, the op amp can perform addition

$$V_o = -\left[\left(\frac{R_3}{R_1}\right)V_1 + \left(\frac{R_3}{R_2}\right)V_2\right]$$

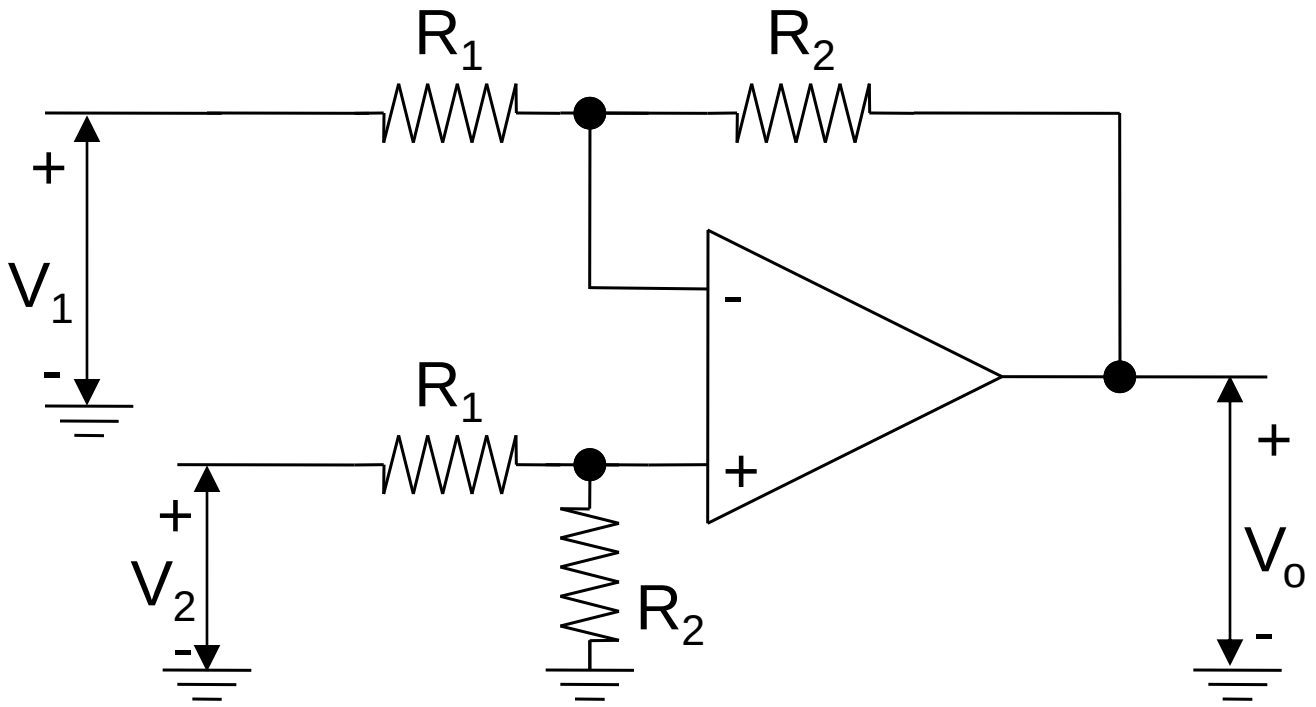


- **The output voltage is a weighted sum of the inputs. For this reason, the circuit in Fig is called a summer.**
- **Needless to say, the summer can have more than three inputs.**

Operational Amplifier for Signal Conditioning

5. Difference Amplifier :

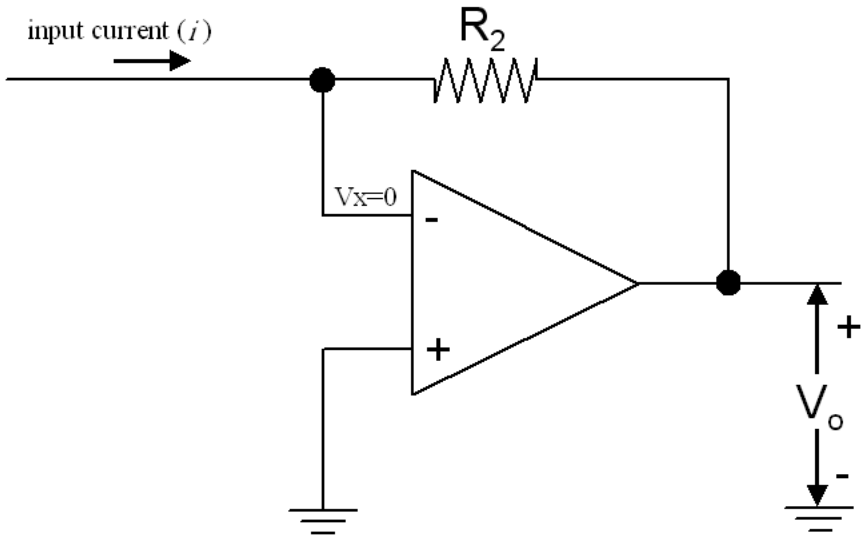
$$V_o = \frac{R_2}{R_1}(V_2 - V_1)$$



Operational Amplifier for Signal Conditioning

6. Current to Voltage Converter:

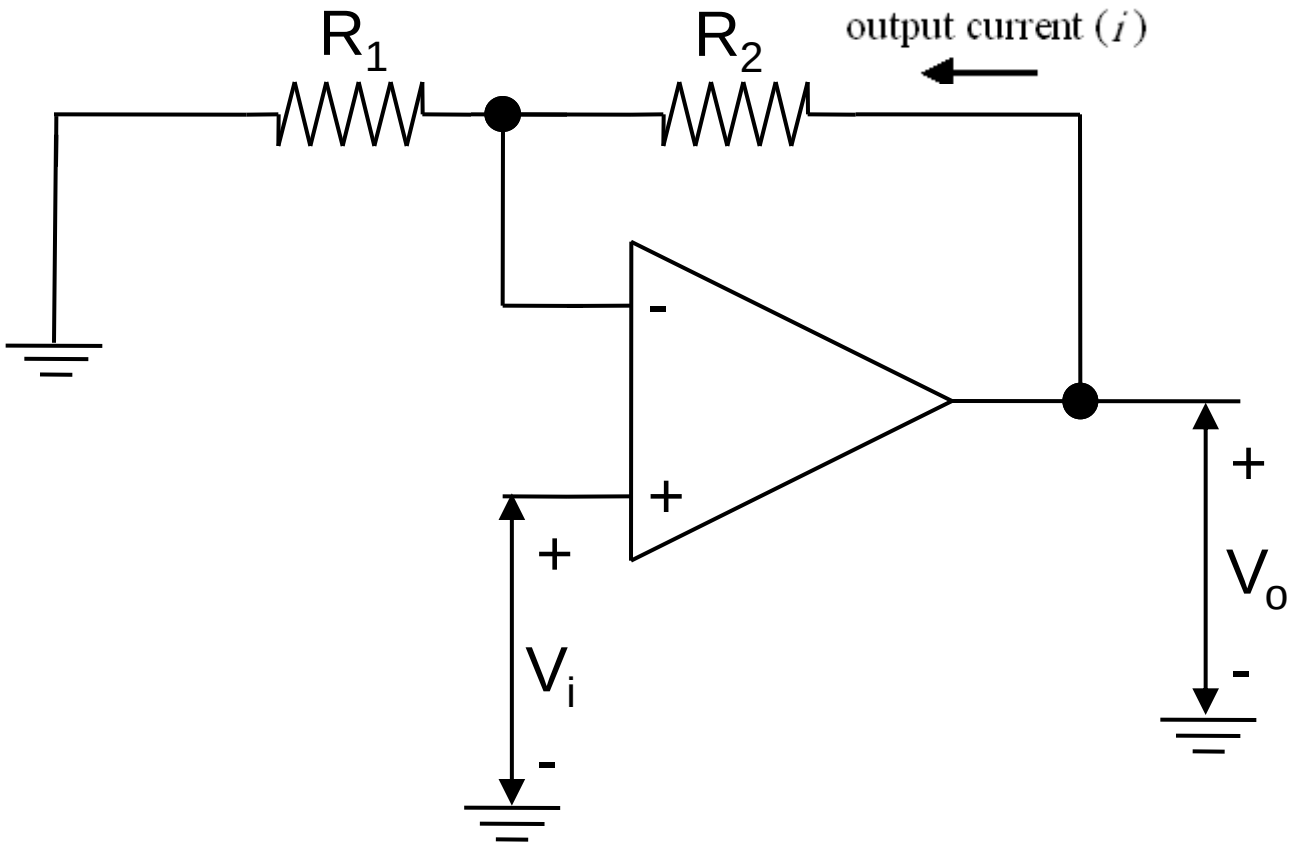
$$V_o = -iR_2$$



Operational Amplifier for Signal Conditioning

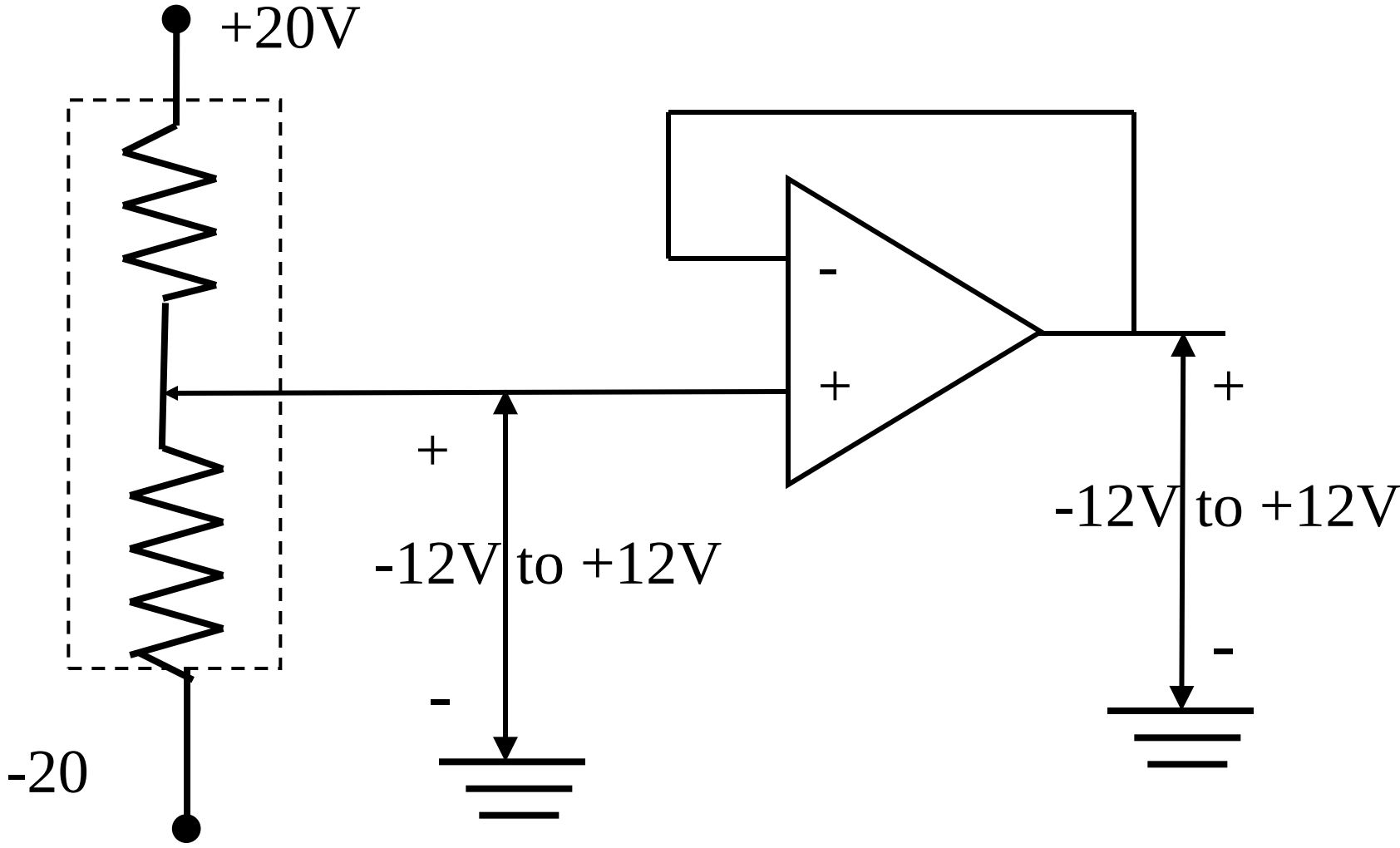
7. Voltage to Current Converter:

$$i = -\frac{V_i}{R_1}$$



Operational Amplifier for Signal Conditioning

8. Voltage Divider “Buffered” :



Operational Amplifier for Signal Conditioning

9. Integrating Amplifier:

$$Ir = Ic$$

$$Vc = Va - V0$$

$$\text{then} \quad \frac{vi - va}{R} = C\frac{dvc}{dt} = c\frac{d(va - v0)}{dt}$$

$$va = vb = 0$$

$$\frac{vi}{R} = -C\frac{dvo}{dt}$$

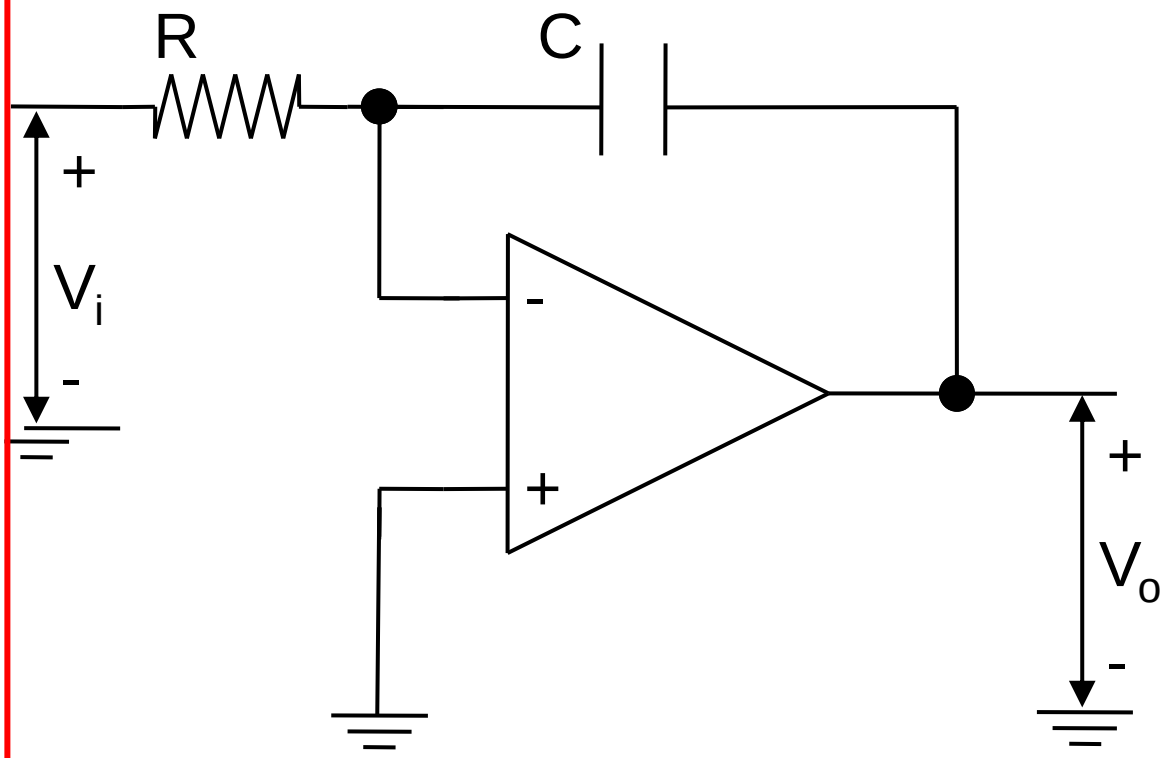
$$dvo = -\frac{1}{RC}vidt$$

$$\int_0^t dv0 = -\frac{1}{RC}\int_0^t vidt$$

$$vo(t) = -\frac{1}{RC}\int_0^t vidt$$

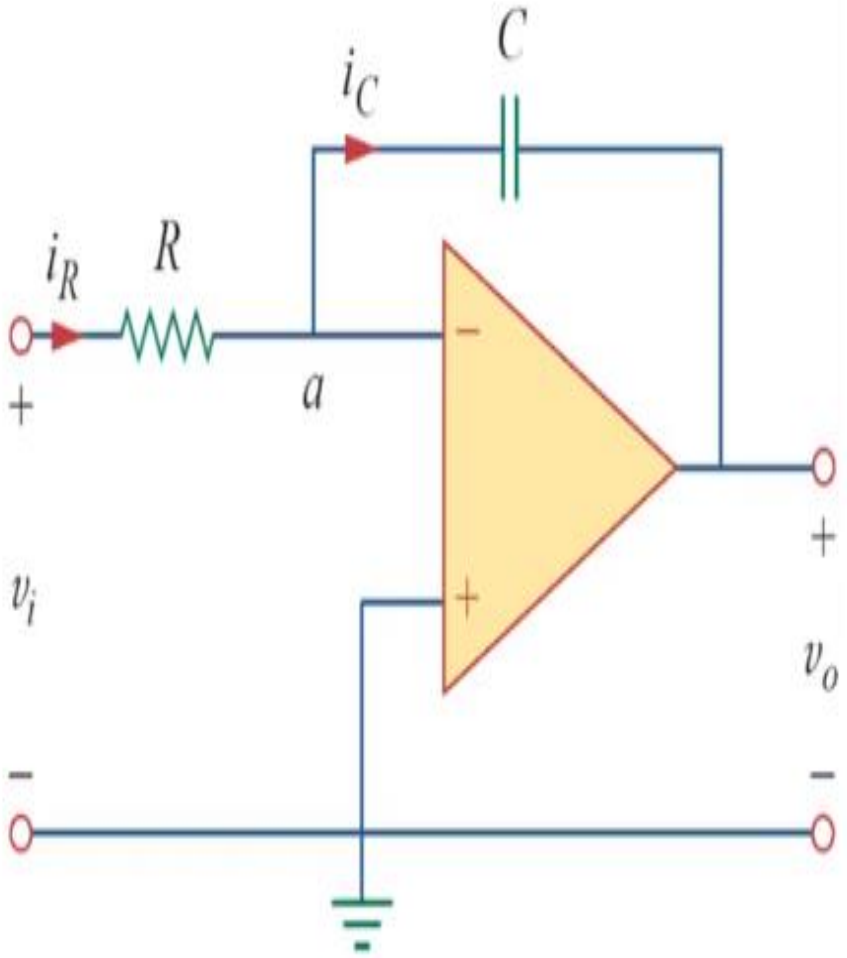
$$V_o = \frac{-1}{RC}\int_0^t V_i dt + V_o(0)$$

$$V_o = \frac{-1}{RC}\int_0^t V_i dt + V_o(0)$$



Ex – The integrator in Fig. 6.35(b) has $R = 100\ \text{k}\Omega$, $C = 20\ \mu\text{F}$. Determine the output voltage when a dc voltage of 10 mV is applied at $t = 0$. Assume that the op amp is initially nulled.

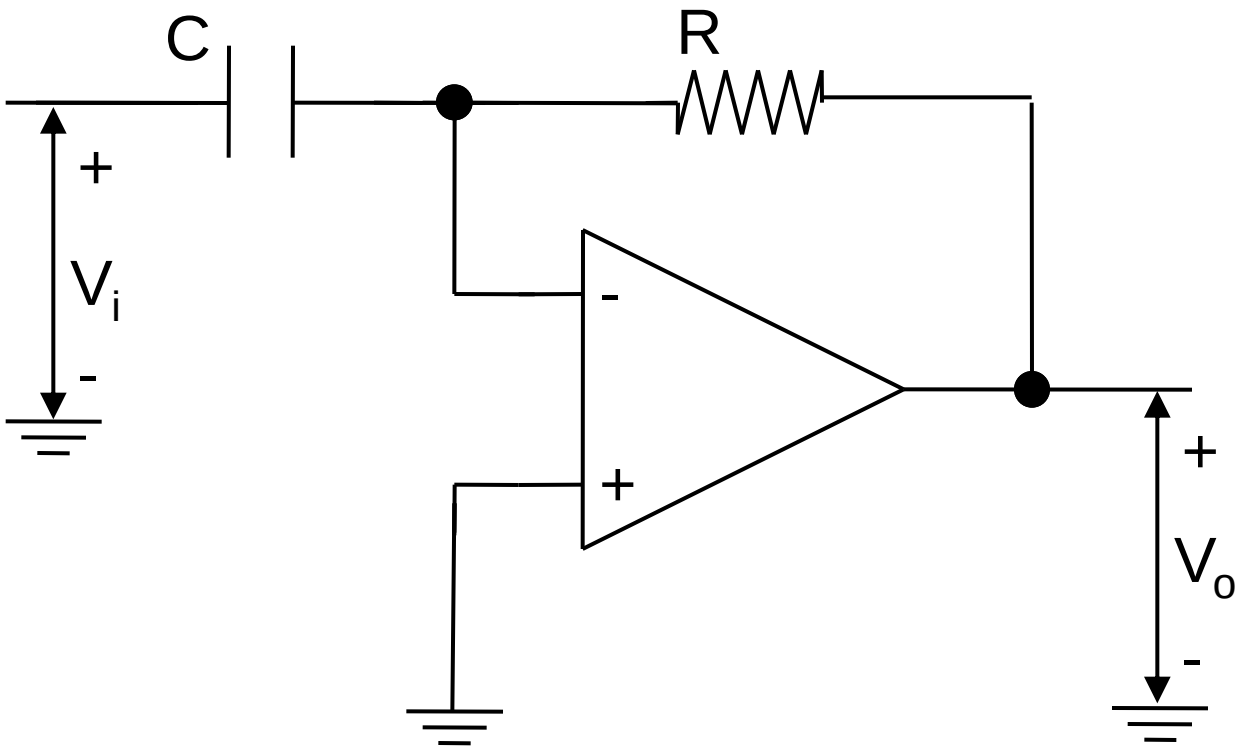
$$v_o(t) - v_o(0) = -\frac{1}{RC}\int_0^t v_i\,dt$$



Operational Amplifier for Signal Conditioning

10. Differentiating Amplifier:

$$V_o = -RC\frac{dV_i}{dt}$$



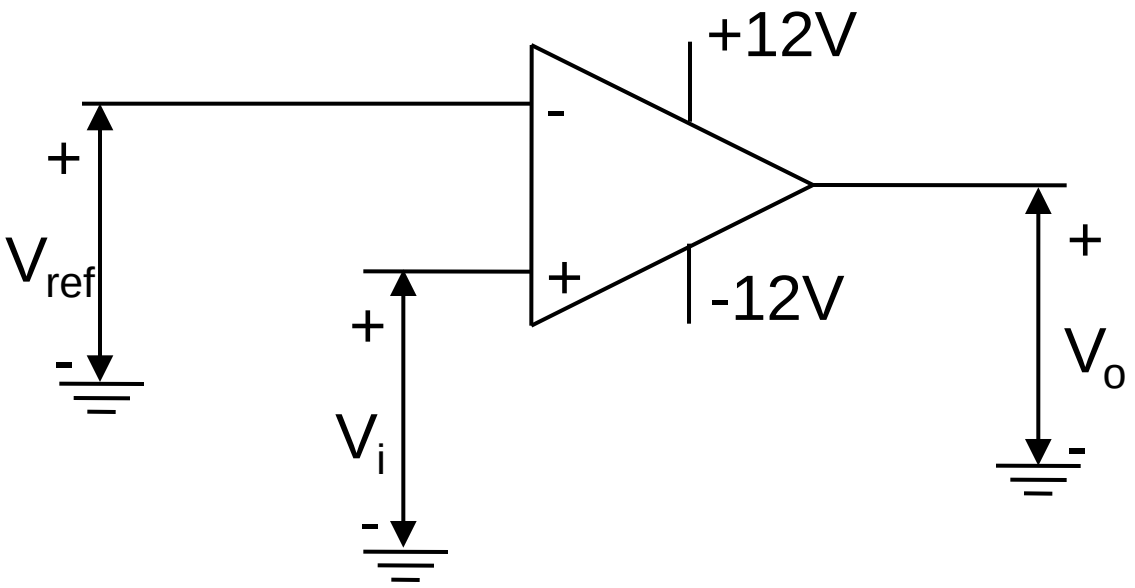
Operational Amplifier for Signal Conditioning

9.Comparator

Logic 1= +12 v
Logic 0 =-12 v
Or
Logic 1 = + 3.2 v
Logic 0 =-0.6 v

$$V_o = \begin{cases} +12V & \text{if } V_i > V_{ref} \\ -12V & \text{if } V_i < V_{ref} \end{cases} \{741,351$$

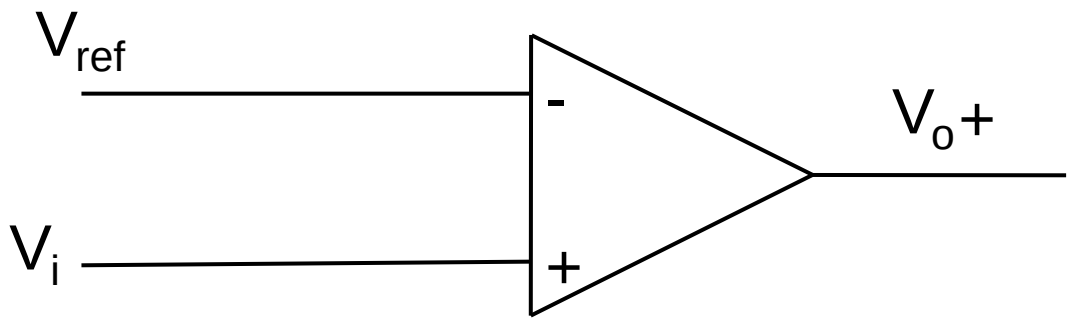
$$V_o = \begin{cases} 3.2\,V & \text{if } V_i > V_{ref} \\ -0.6\,V & \text{if } V_i < V_{ref} \end{cases} \{710$$



EX Operational Amplifier for Signal Conditioning

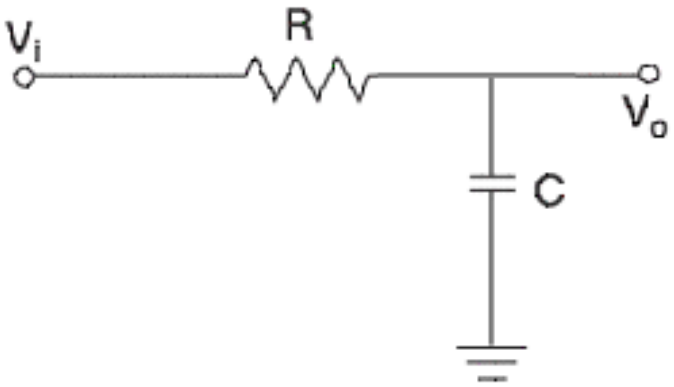
EX: find Vo(t) assuming 710 comparator if

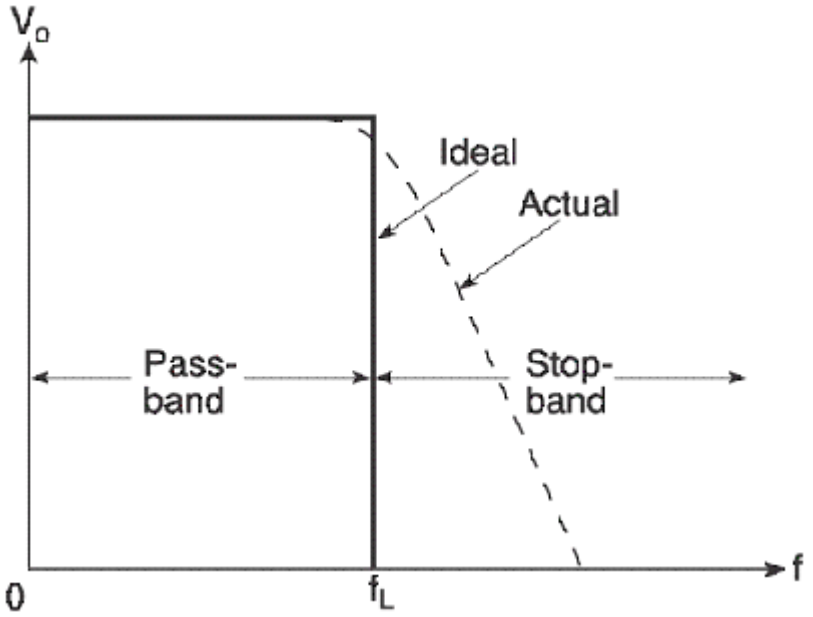
Vi (t)= 10 sinwt Vref= 7v



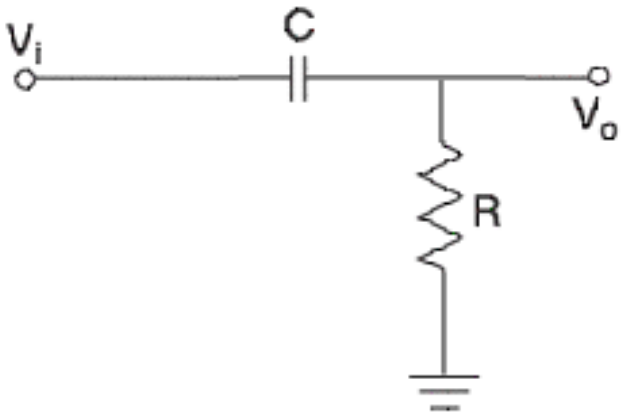
Filters

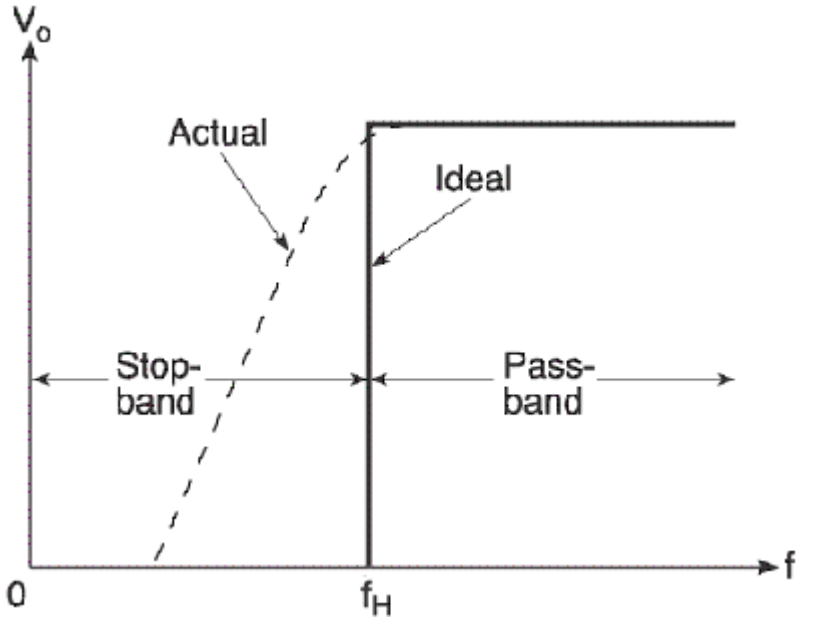
1.Low pass filter:

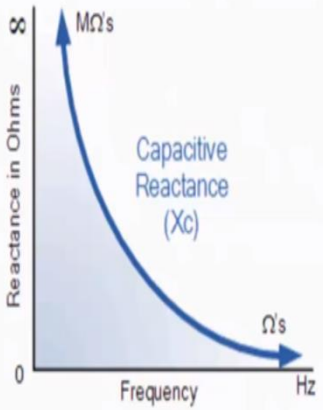


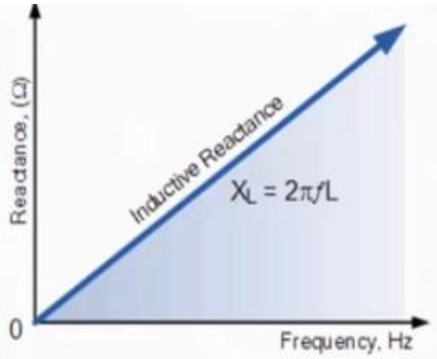


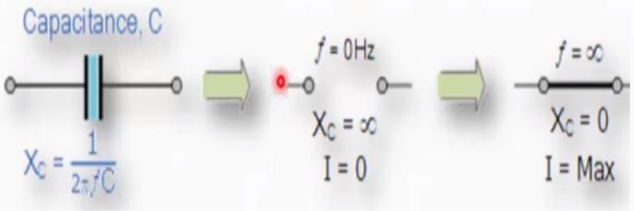
2.High pass filter

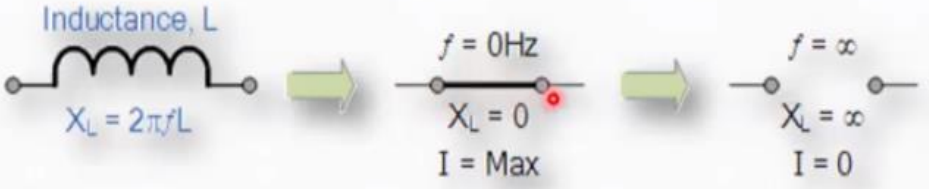




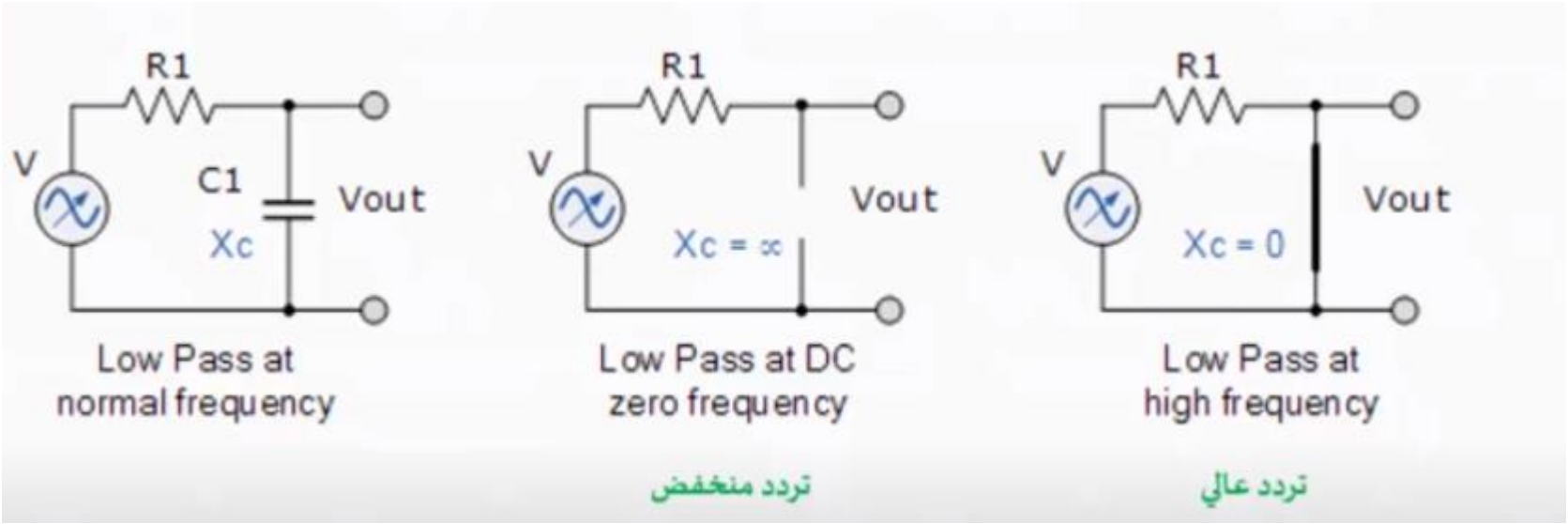


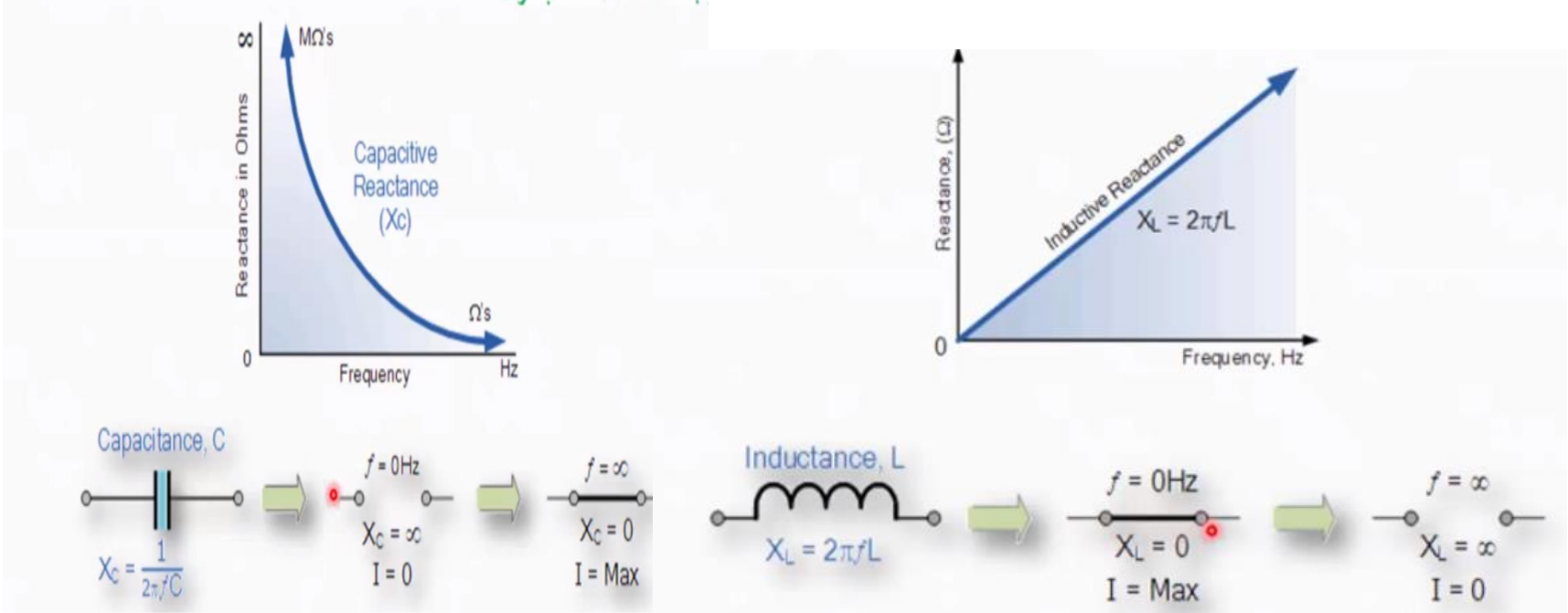




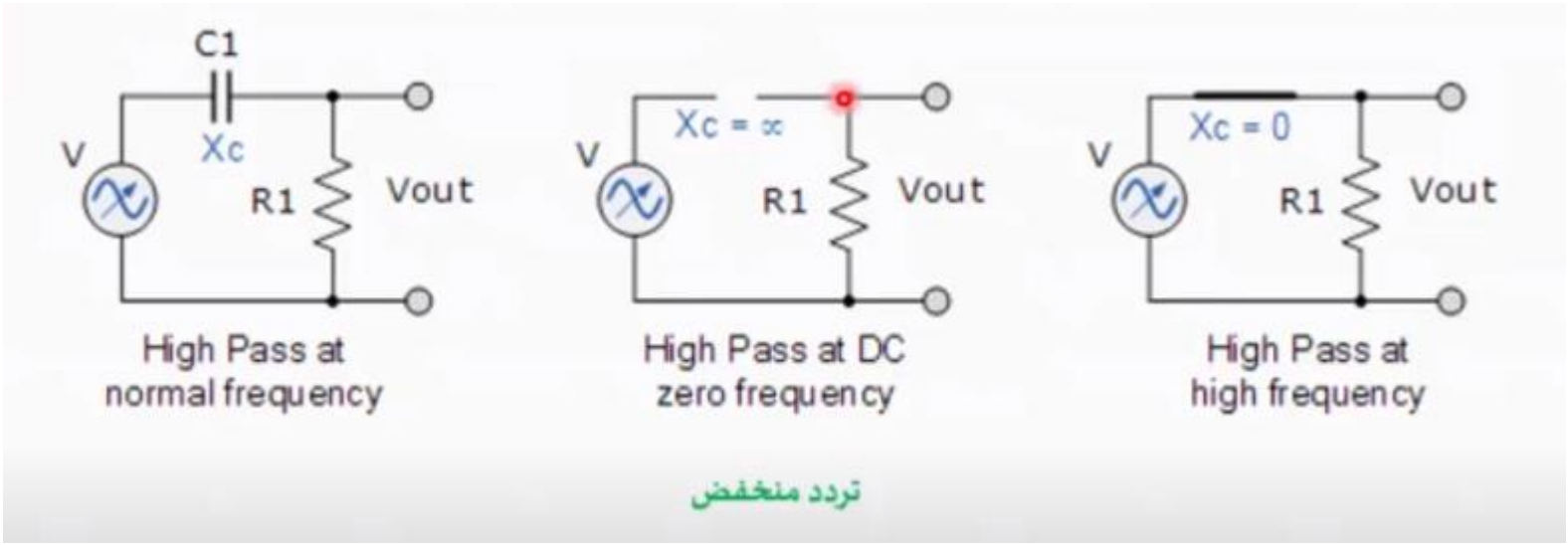


Low pass filter:



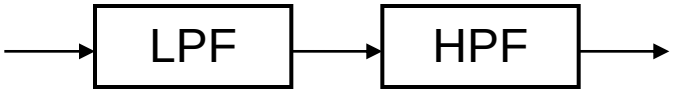


High pass filter:



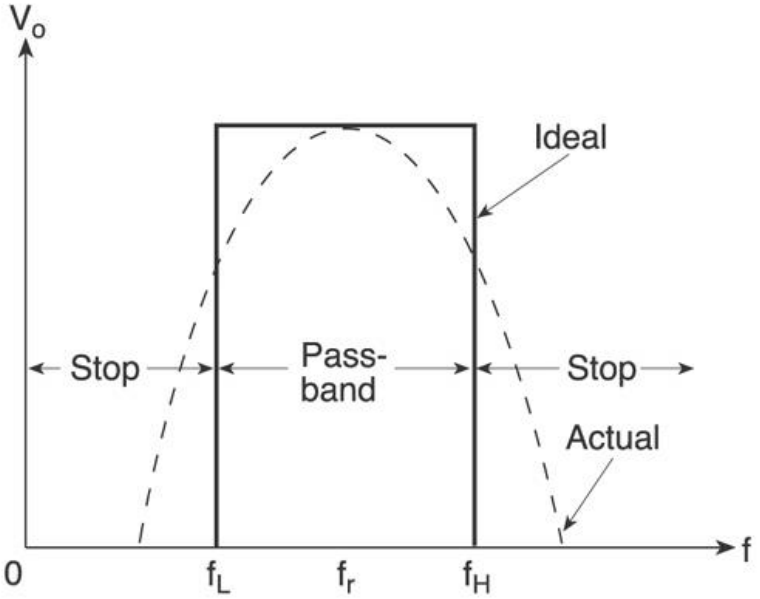
Filters

3.Band pass filter:

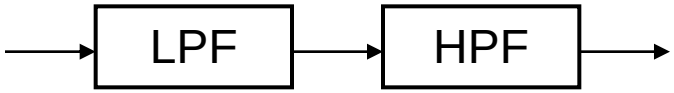


$f_L > f_H$

$BW = f_H - f_L$

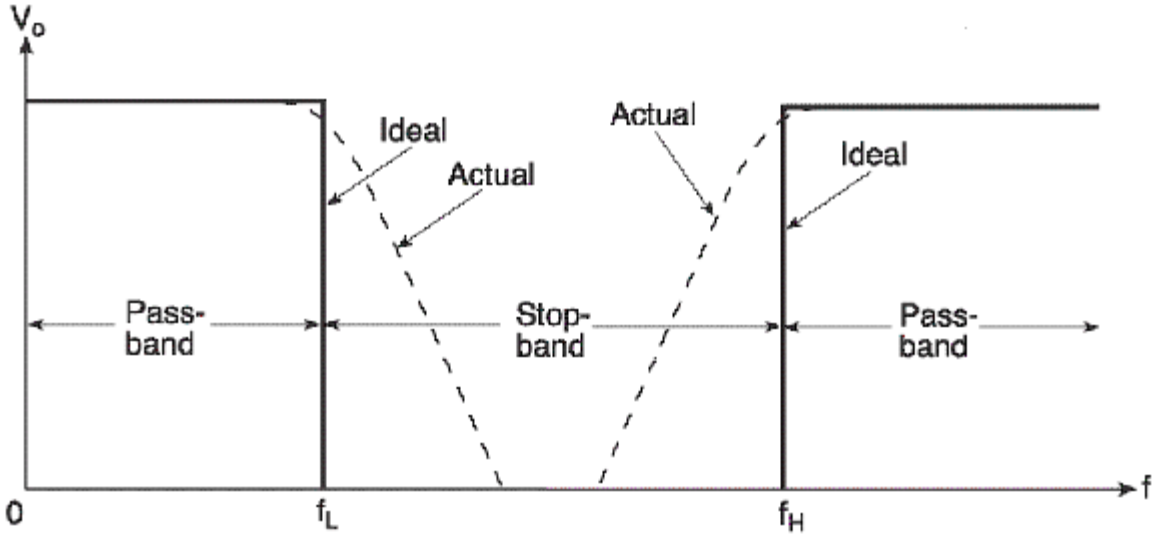


4.Band reject filter



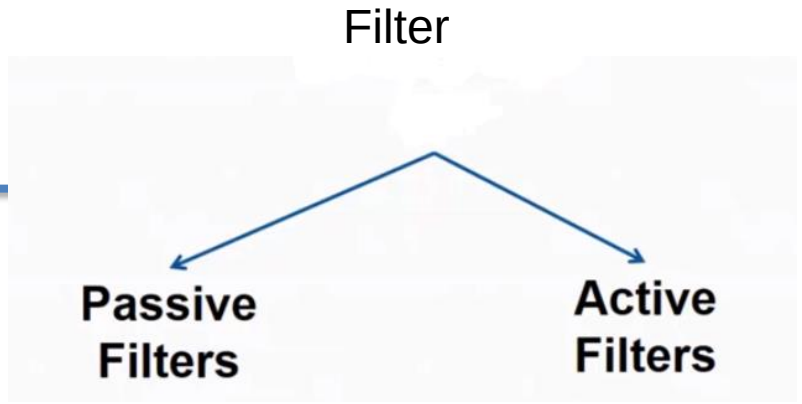
$f_L < f_H$

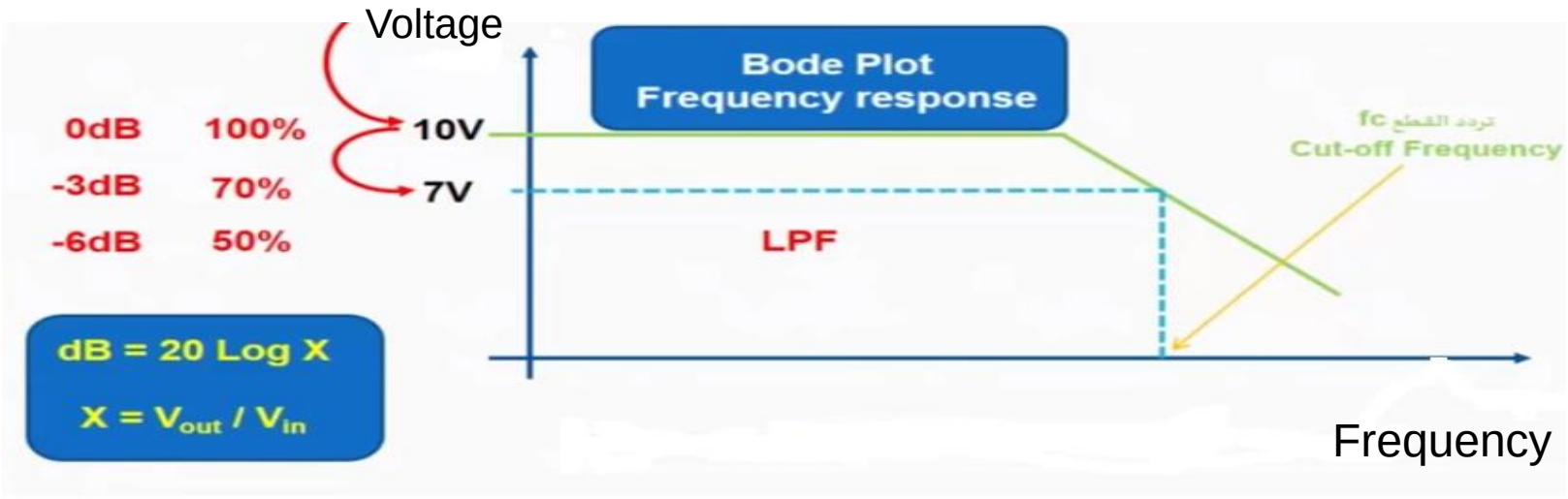
$RBW = f_H - f_L$



Filters

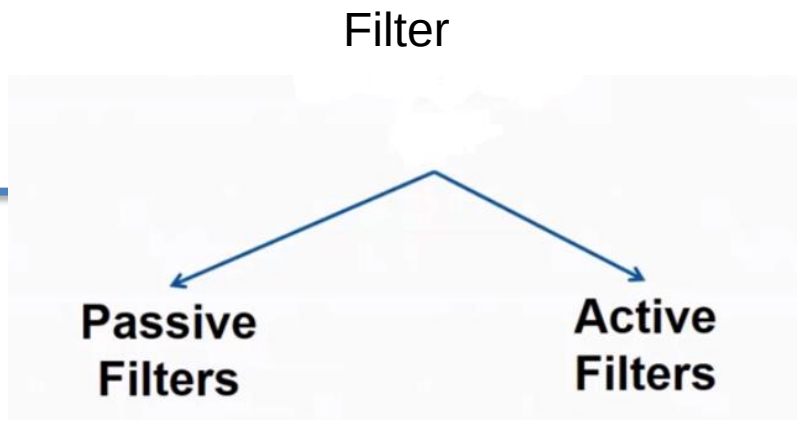
a **filter** is a device or process that removes some unwanted components or features from a signal

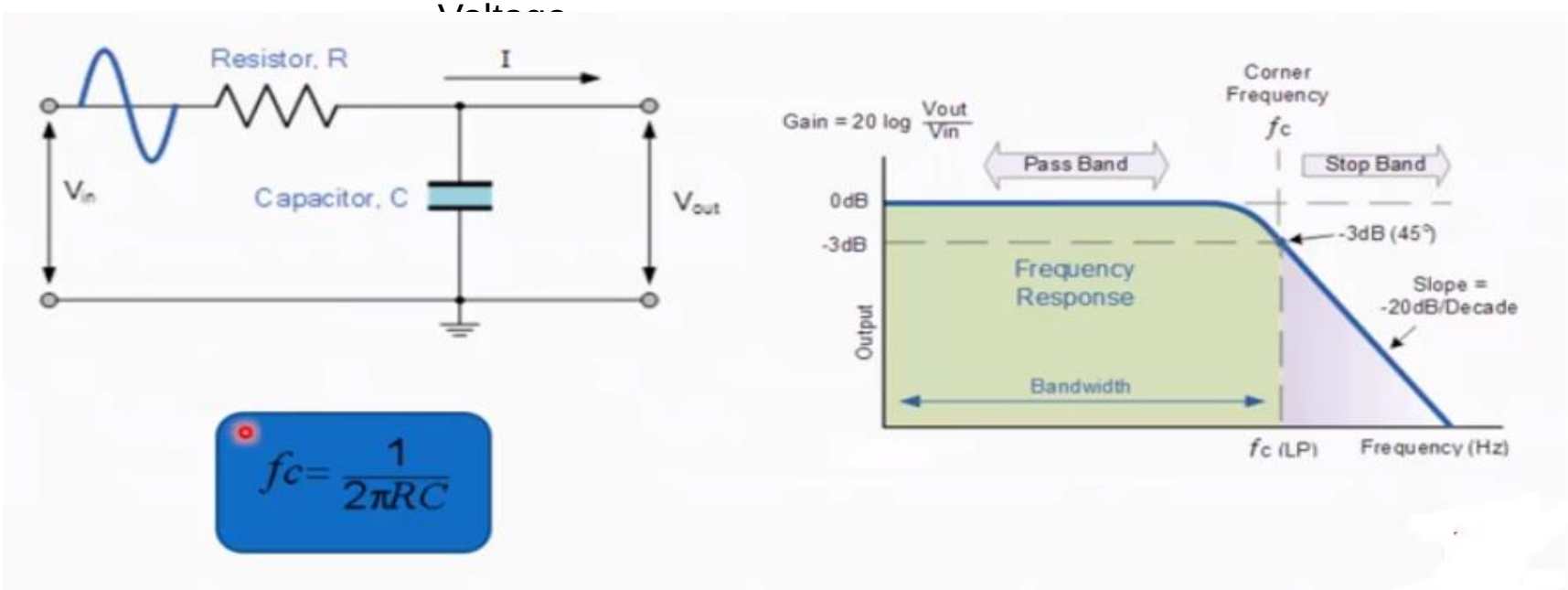




Filters

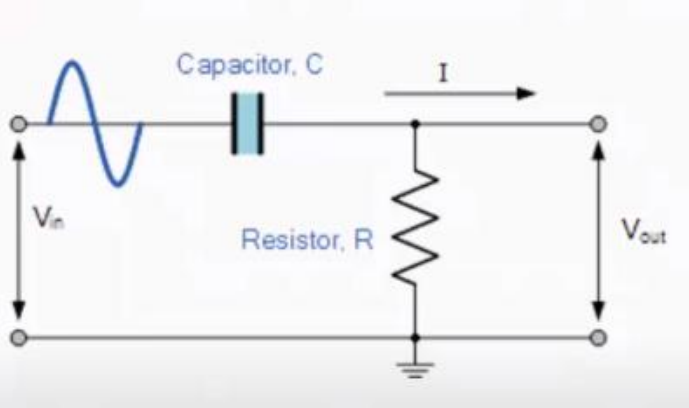
a **filter** is a device or process that removes some unwanted components or features from a signal





Example

- Find the cut frequency in high pass filters if R =240 KΩ and C =83pF



Real time system
Lecture 4: analog to digital converters (ADC)

ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.

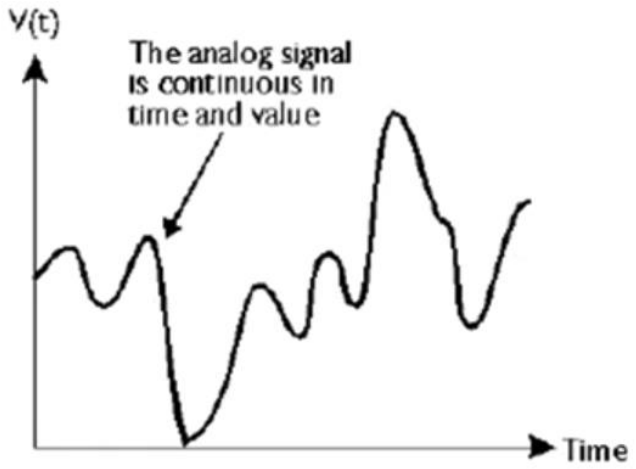
Introduction of ADC

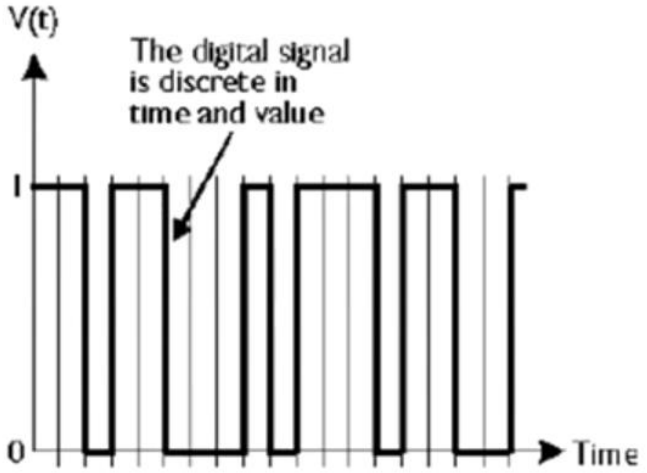
- What is ADC?
- Why is ADC important?
- How does it work?
- **Type of ADC**

Introduction

- Analog to Digital Converters(ADCs) are used to convert analog signals to digital signals.
- The conversion of an analog (continuous) voltage x(t) into a discrete sequence of numbers x(n) is performed by (ADC) in order to be processed by digital computer.

- The ADC works by **sampling** the amplitude of the analog signal at regular intervals in time, and **encodes** (quantizes) those values as binary numbers.





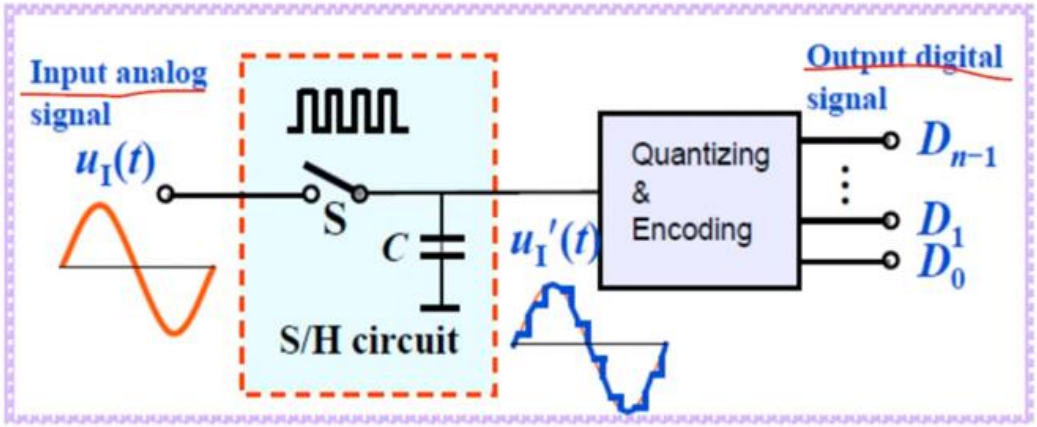
Why is ADC Important?

- All microcontrollers store information using digital logic
- Compress information to digital form for efficient storage
- Medium for storing digital data is more robust
- Digital data transfer is more efficient
- Digital data is easily reproducible
- Provides a link between real-world signals and data storage

How ADC Works

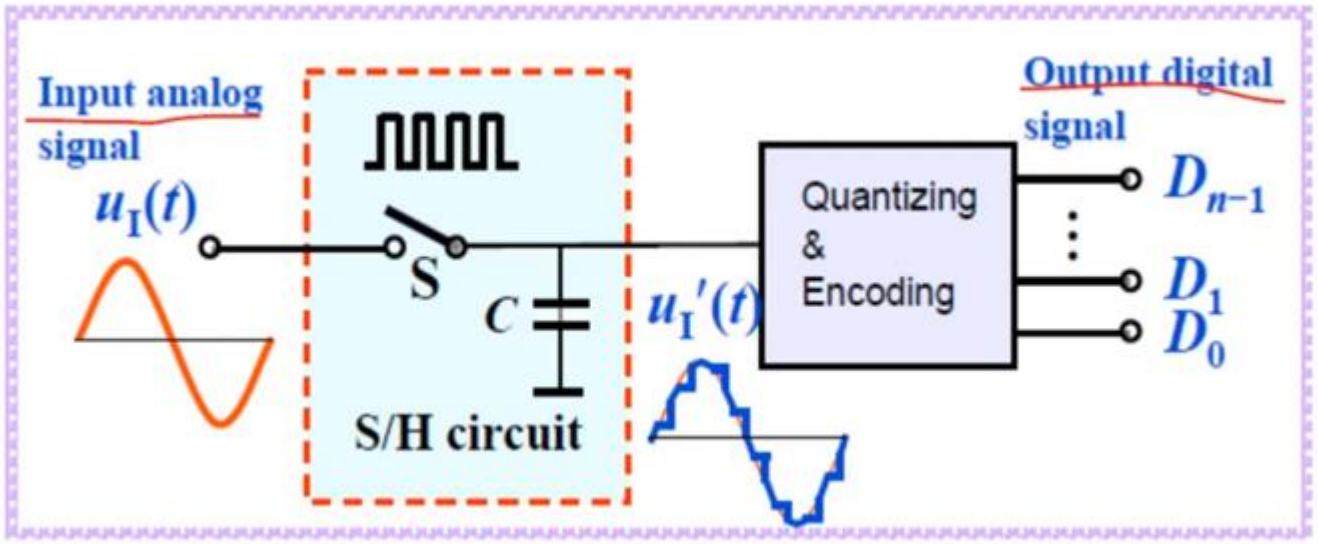
2 Stages:

- Sampling
  - Sample-Hold Circuit
  - Aliasing
- Quantizing and Encoding
  - Resolution



Sampling

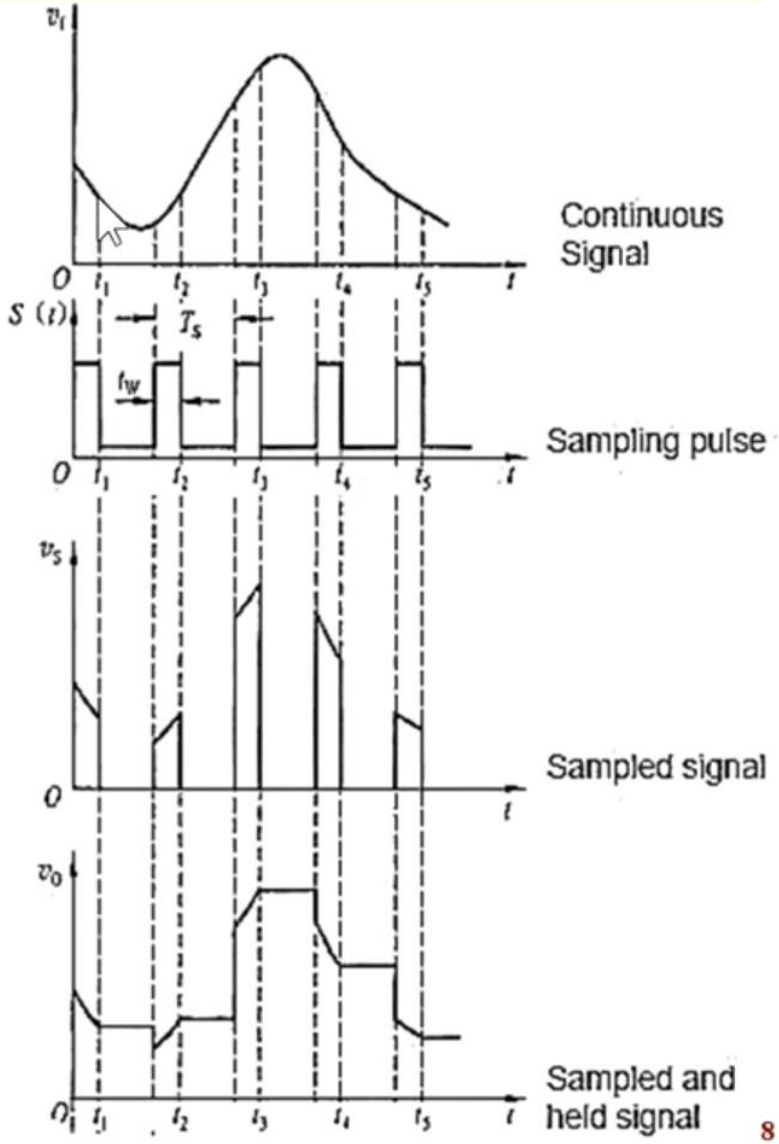
- Reduction of a continuous signal to a discrete signal
- Achieved through sampling and holding circuit
- Switch ON – sampling of signal (time to charge capacitor w/ V_{in})
- Switch OFF - voltage stored in capacitor (hold operation)
- Must hold sampled value constant for digital conversion



ADC process

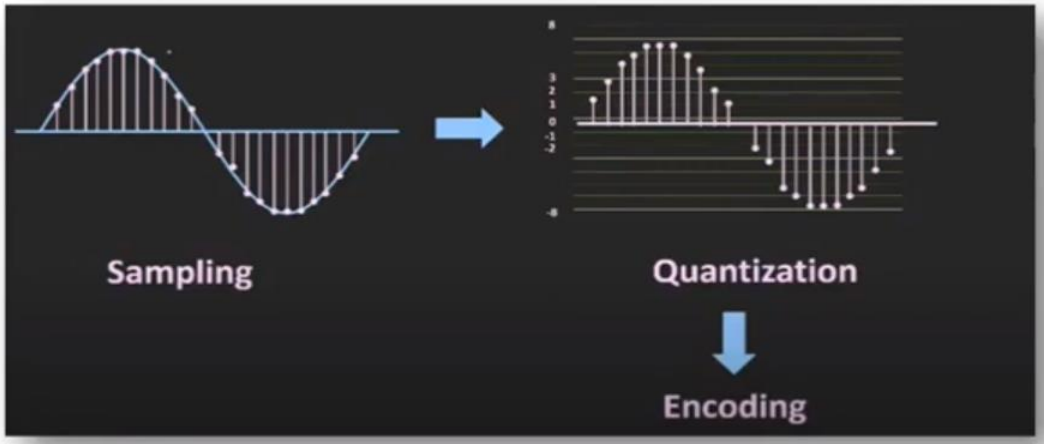
Sampling and Holding:

- It is a process of taking a sufficient number of discrete values at point on a waveform that will define the shape of waveform.
- The more samples you take, the more accurately you will define the waveform.
- The regular time intervals are known as the sampling period (Ts) and are determined by the ADC clock.
- This period defines the frequency at which the sampling will be done, such that the sampling frequency (in Hertz) is: **fs =1/Ts**
- The goal is to convert analog signal into series of impulses, each representing amplitude of the signal at given point as shown in the Figure.



ADC process

- Quantization: Is the process of converting the sampled continuous signals into discrete-valued data (set of finite states).



- Resolution is the minimum change that can be detected by the ADC
- Resolution depends on number of bits

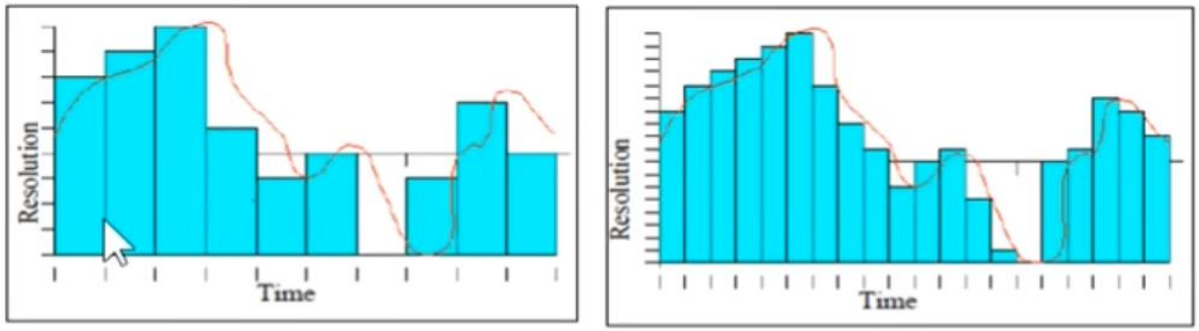
R= Vi max – Vi min / 2^no. of bits

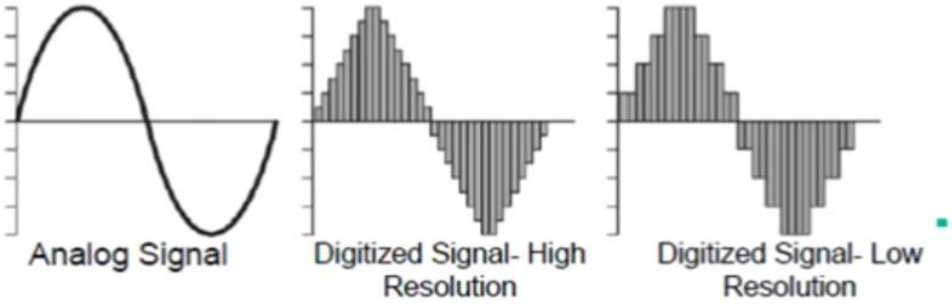
Accuracy of ADC

There are two ways to best improve accuracy of A/D conversion:

- Increasing the resolution which improves the accuracy in measuring the amplitude of the analog signal.
- Increasing the sampling rate which increases the maximum frequency that can be measured.

- Low Accuracy
- Improved





Accuracy of ADC

Example: if you have 0-10V signal. What is the Discrete Voltage Ranges and Output Binary Equivalent by using 3 bit A/D converter?

Solution:-

$N=2^n$

For a 3 bit A/D converter, $N=2^3=8$. Analog quantization size:

$R = (V_{max} - V_{min}) / N = (10V - 0V) / 8 = 1.25V$

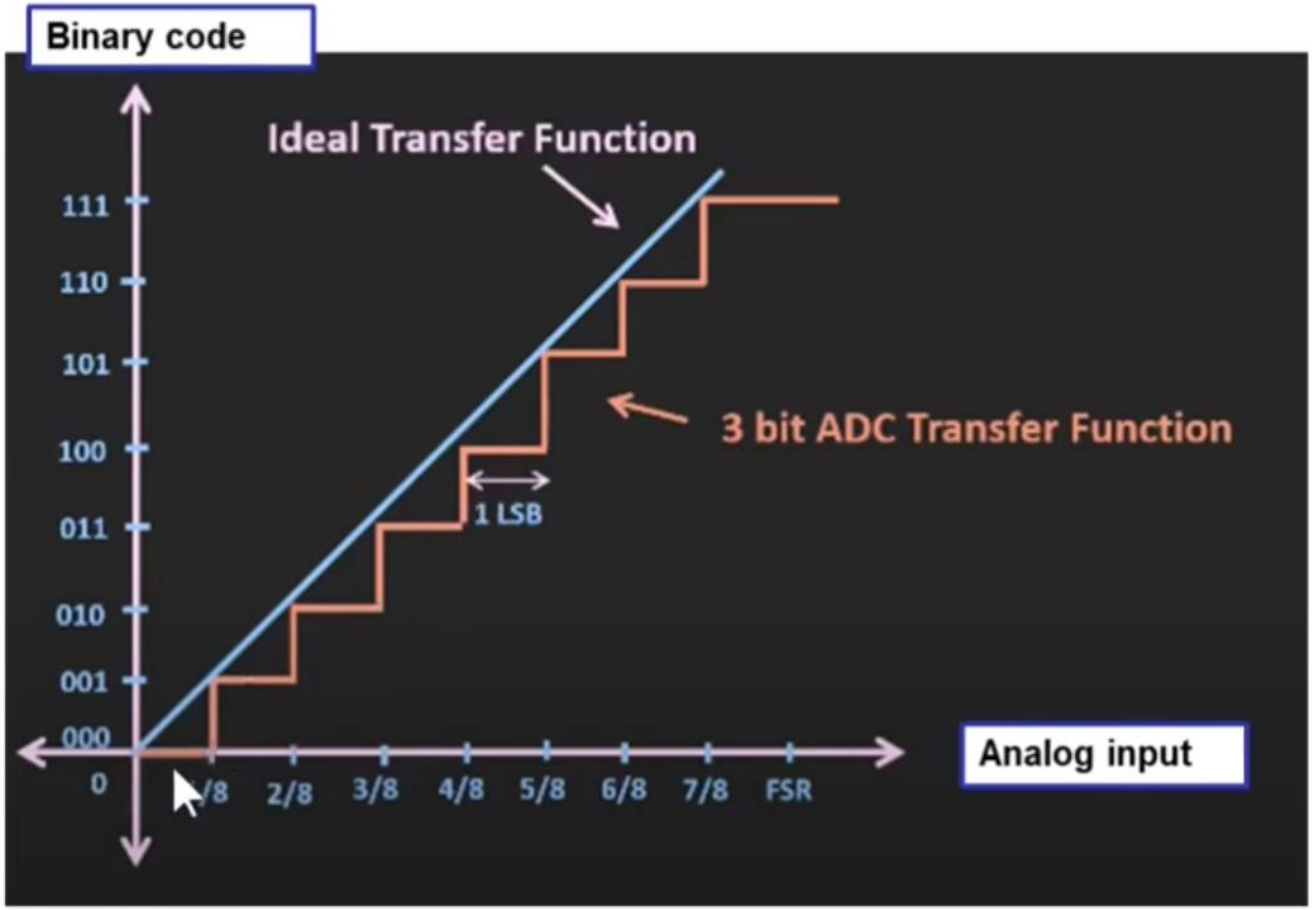
Step 1: Quantizing:

Output States	Discrete Voltage Ranges (V)
0	0.00-1.25
1	1.25-2.50
2	2.50-3.75
3	3.75-5.00
4	5.00-6.25
5	6.25-7.50
6	7.50-8.75
7	8.75-10.0

Step 2: Encoding: Here we assign the digital value (binary number) to each state for the computer to read.

Output States	Output Binary Equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Accuracy of ADC



Accuracy of ADC

- ***Reference Voltage***
- The reference voltage is the maximum value that the ADC can convert.
- An 8bit ADC can convert values from 0V to the reference voltage. This voltage range is divided into 256 values, or steps. The size of the step is given by the following equation:

$$\frac{\text{Reference Voltage}}{256} = \frac{5\text{V}}{256} = .0195\text{V, or } 19.5\text{mV} \quad = 1\text{LSB}$$

- This is the step size of the converter. It also defines the converter's resolution.
- Note that ADC or DAC cannot be more accurate than its reference.

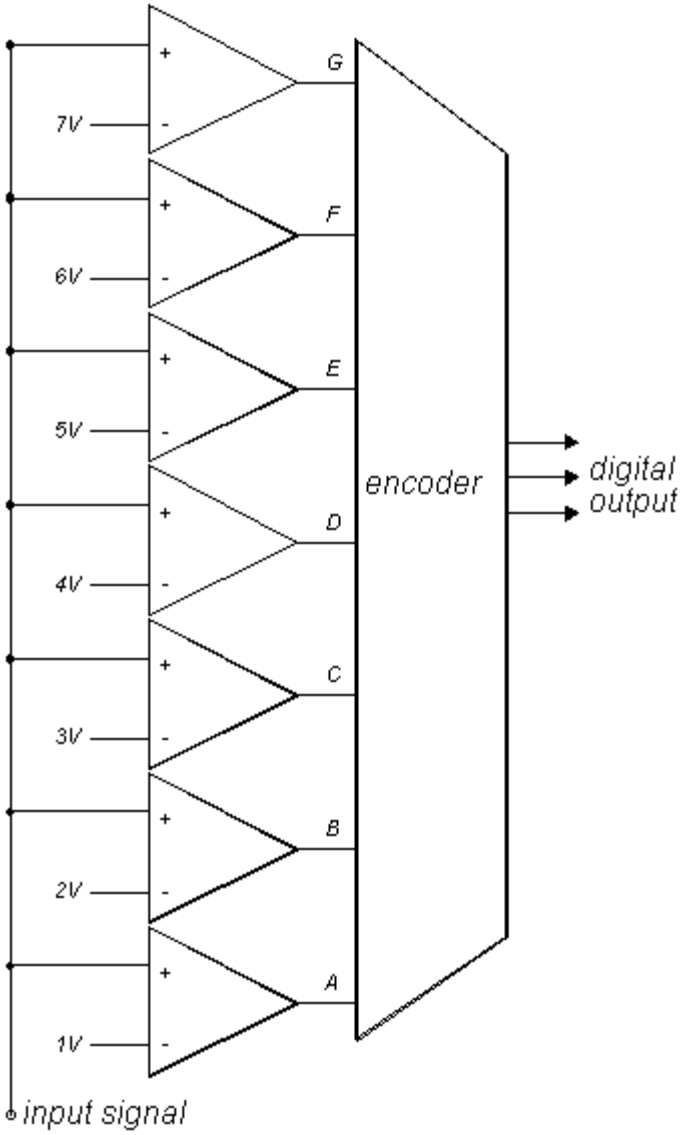
Types of A/D Converters

- Flash A/D Converter
- Successive Approximation A/D Converter
- Example of Successive Approximation
- Dual Slope A/D Converter
- Delta – Sigma A/D Converter

Analogue to Digital Converter

1. Flash A to D Converter:

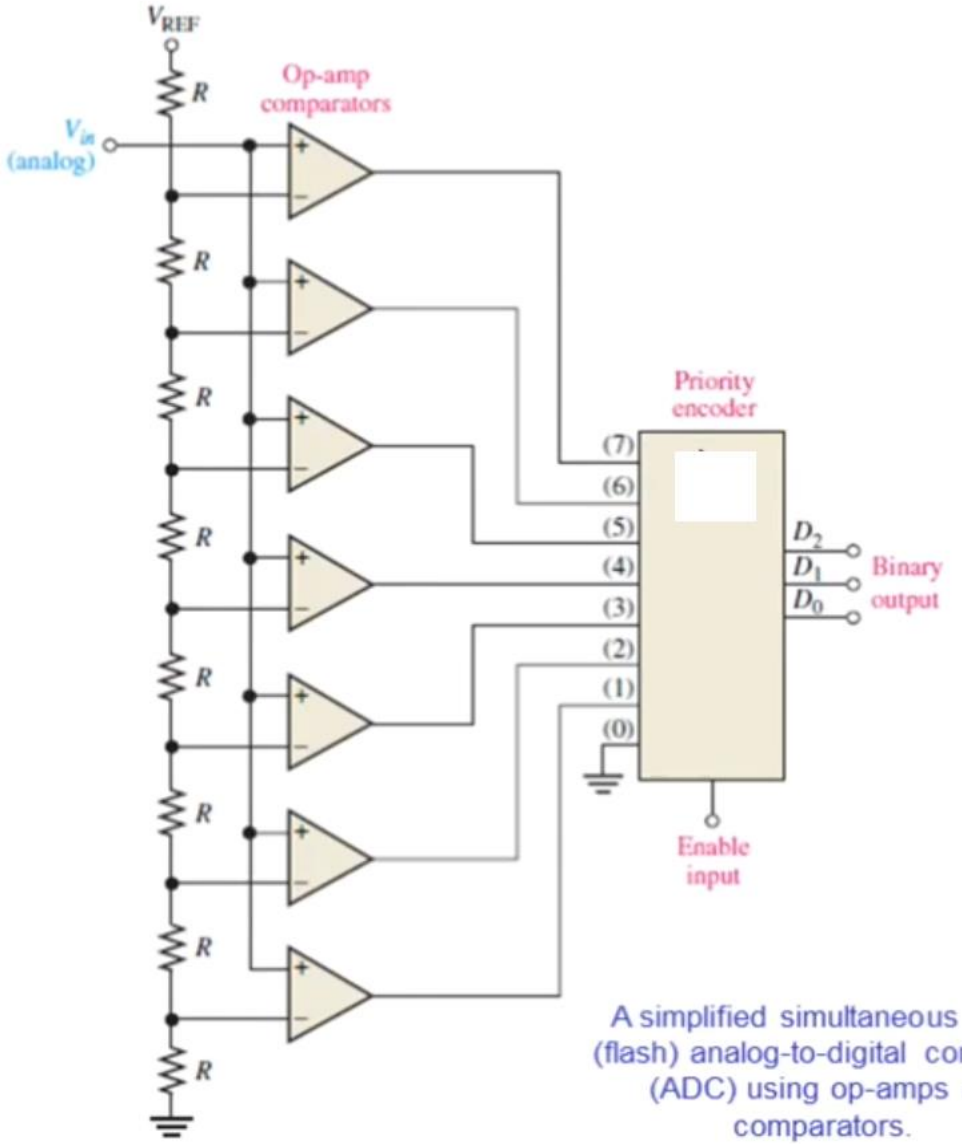
- Uses a reference and a comparator for each of the discrete levels represented in the digital output
- Not practical for more than 10 bit converters generally fast but expensive



- The analog-to-digital converter (ADC) shown produces three-digit binary numbers on its output, which represent the values of the analog input voltage as it changes.

- This converter requires seven comparators. In general, $2^n - 1$ comparators are required for conversion to an n-digit binary number.

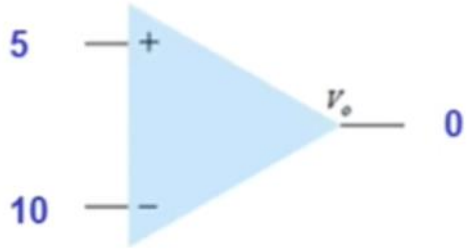
- This ADCs are useful in applications that require the fastest possible conversion times, such as video processing.



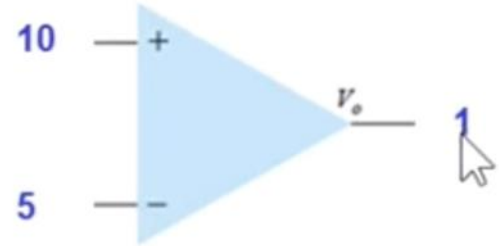
A simplified simultaneous 3 bit (flash) analog-to-digital converter (ADC) using op-amps as comparators.

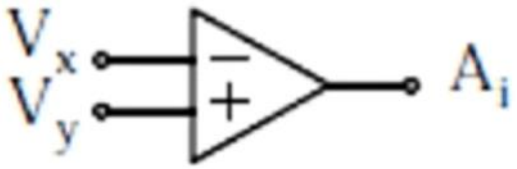
Comparator compares two voltages value and gives output high (1) or low (0)

- In this comparator the inverting terminal (-) value is higher than non inverting (+) terminal. So the output is low (0).



- In this comparator the inverting terminal (-) value is lower than non inverting (+) terminal. So the output is high (1).

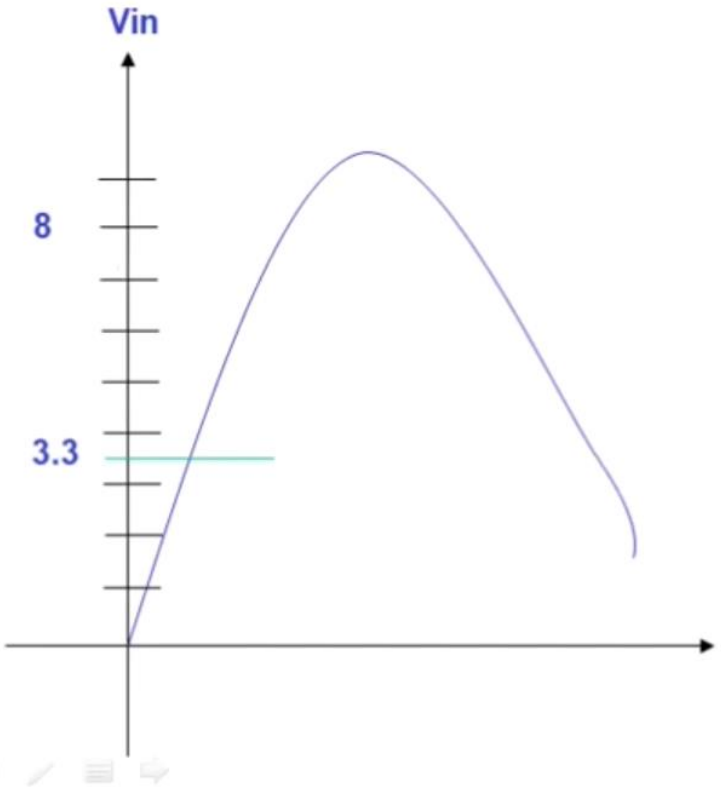


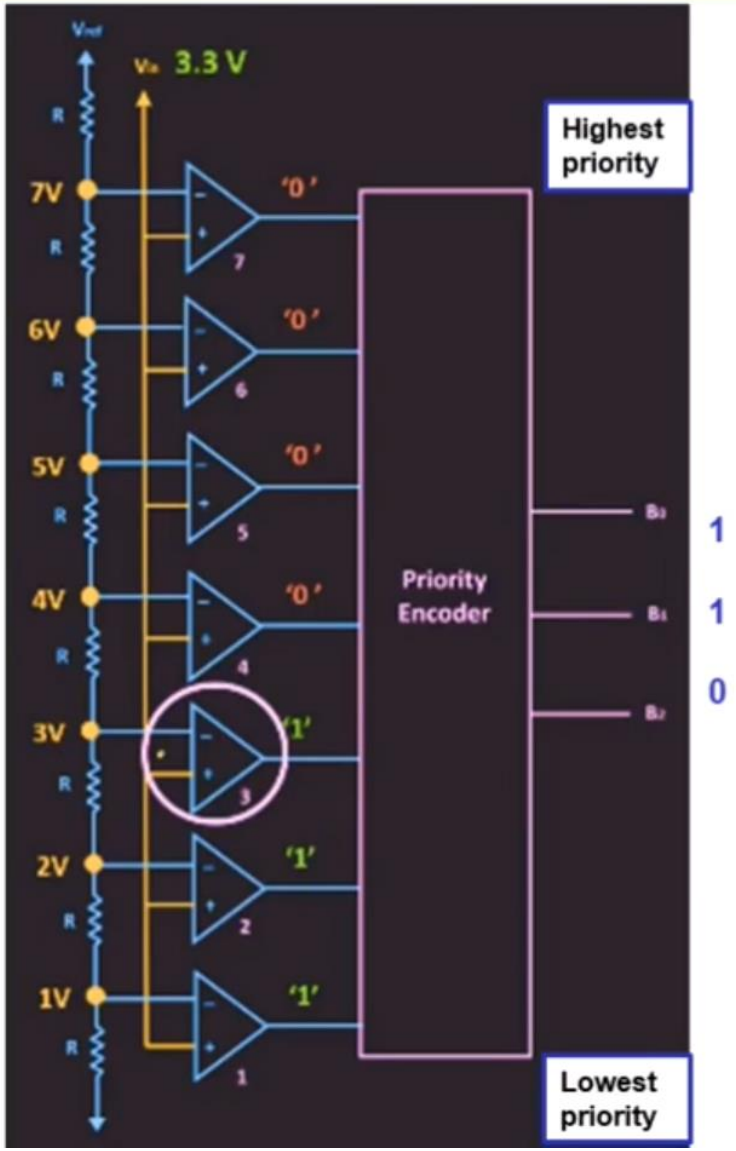


Inputs	Output
$V_x = V_y$	Previous Value
$V_x > V_y$	$A_i = 0$
$V_x < V_y$	$A_i = 1$

Analogue to Digital Converter

- Vref=8v
- Vin=3.3v





Analogue to Digital Converter

Advantages

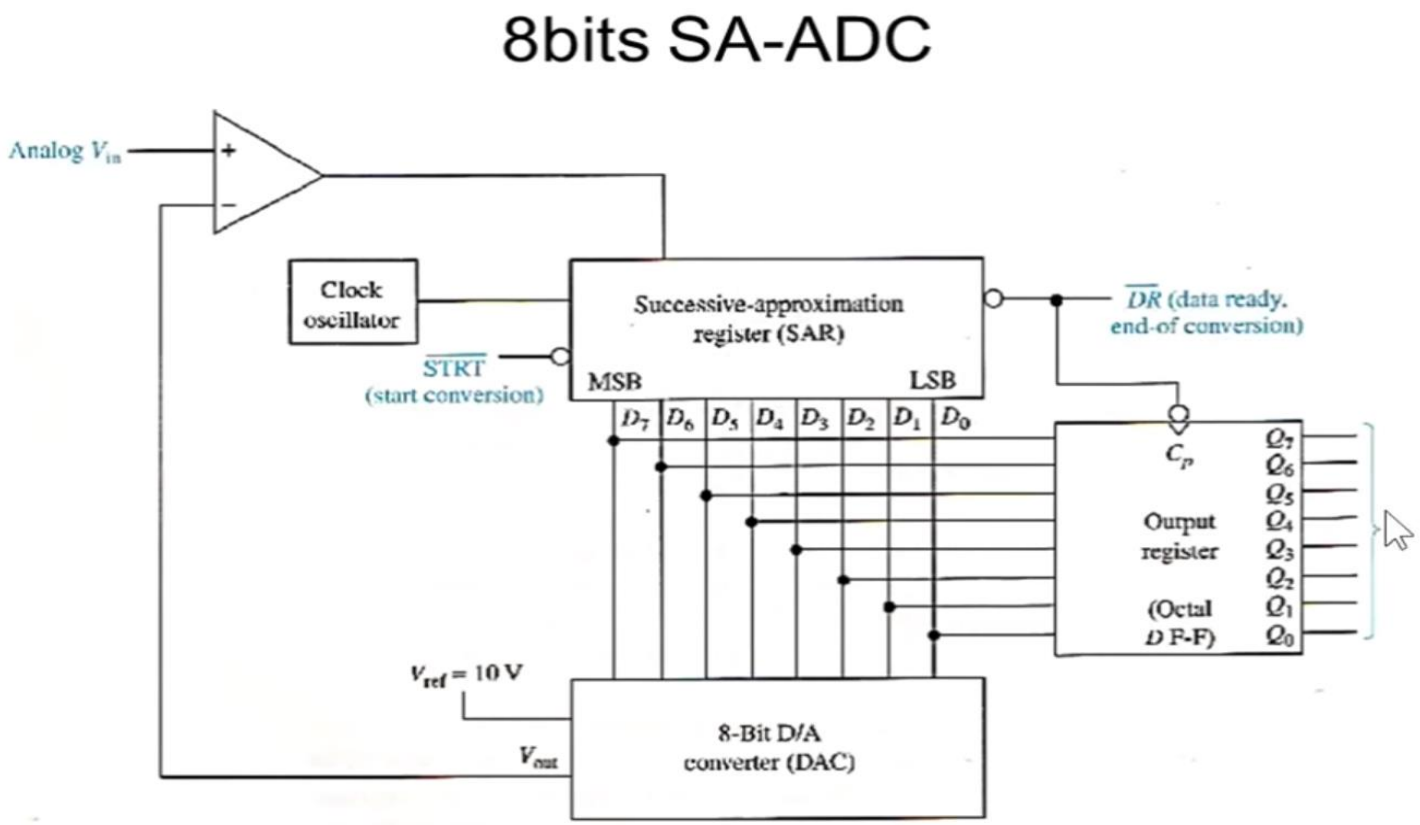
- Simplest in terms of operational theory,
- most efficient in terms of speed (very fast).
- Suitable for large bandwidth applications (Satellite)

Disadvantages

- Lower resolution, Expensive
- for each additional output bit, the number of comparators is increase.
- Component matching problem (comparators & Resistors)

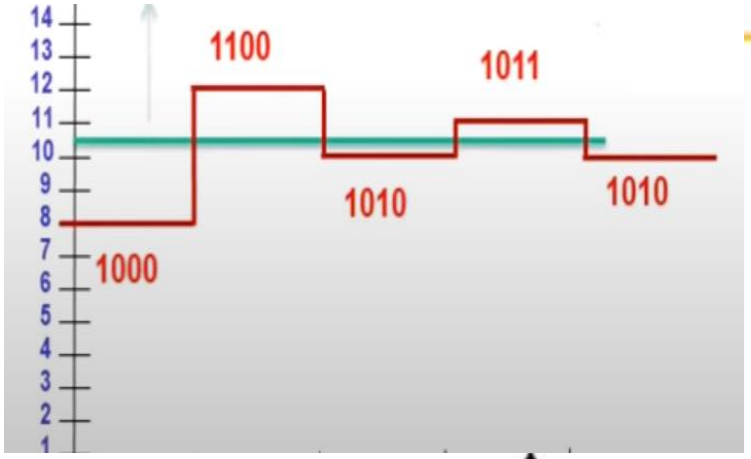
Analogue to Digital Converter

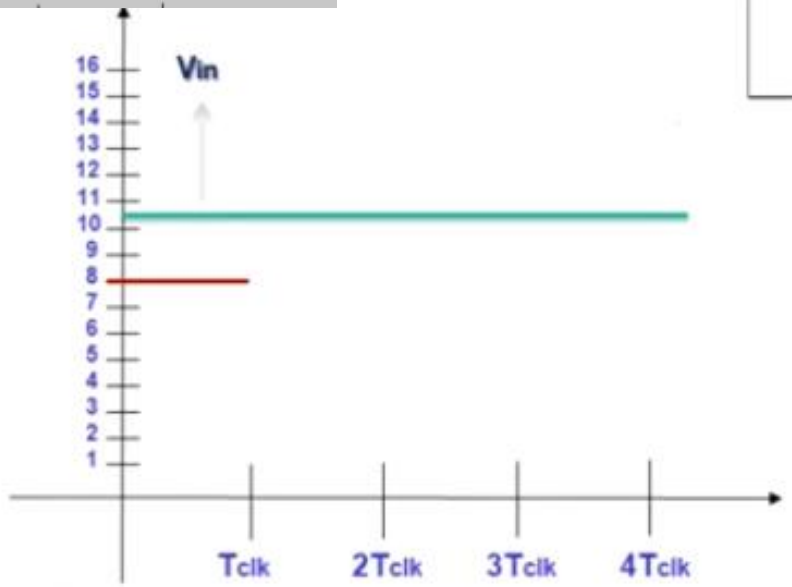
2. Successive-approximation converter: This converter contains an 8-bit successive-approximations register (SAR).



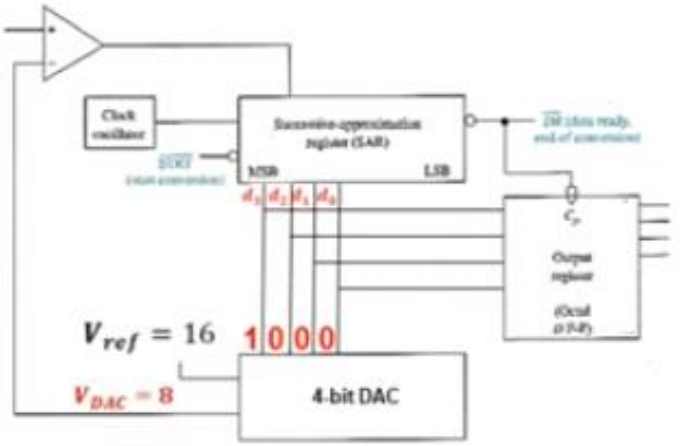
Analogue to Digital Converter

4-bits SA-ADC





$V_{in} = 10.5$



$$V_{DAC} = V_{ref}\left[\frac{d_3}{2} + \frac{d_2}{4} + \frac{d_1}{8} + \frac{d_0}{16}\right]$$

MSB LSB

Example

For 8bits SA-ADC, find the binary output for analog input voltage Van = 198mV.

For 8-bit ADC start with set high the MSB 10000000

1	0	0	0	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Van=198 ≥ 128	10000000	$yes = 1$
Van $\geq 128 + 64$	11000000	$yes = 1$
Van $\geq 128 + 64 + 32$	11100000	$NO = 0$
Van $\geq 128 + 64 + 16$	11010000	$NO = 0$
Van $\geq 128 + 64 + 8$	11001000	$NO = 0$
Van $\geq 128 + 64 + 4$	11000100	$yes = 1$
Van $\geq 128 + 64 + 4 + 2$	11000110	$yes = 1$
Van $\geq 128 + 64 + 4 + 2 + 1$	11000111	$NO = 0$

Binary output = 11000110

Analogue to Digital Converter

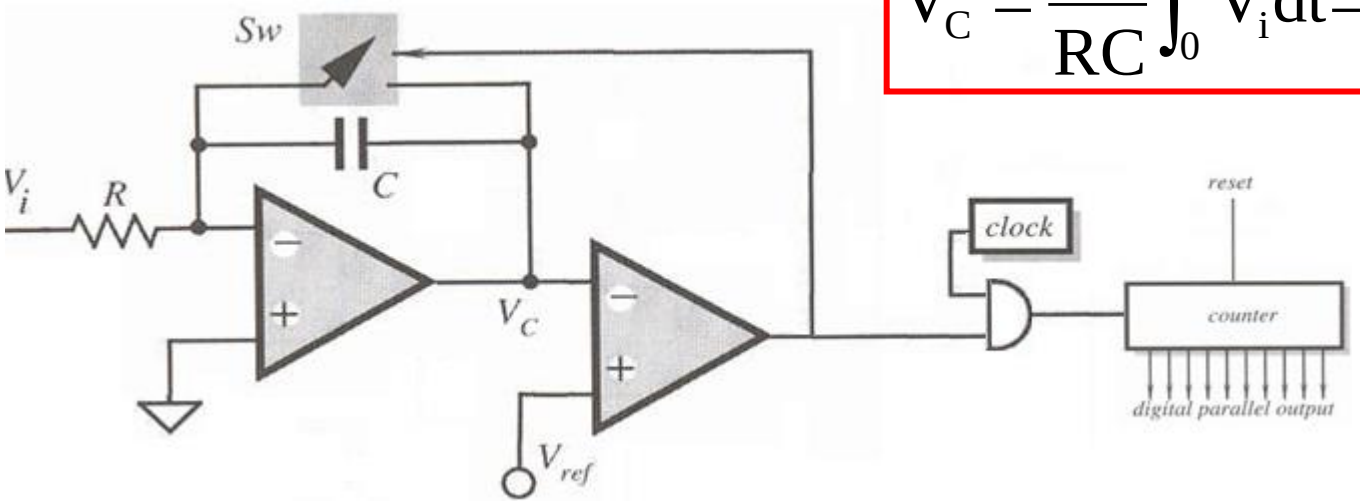
3. Single & Dual-Slope ADC

- Another type of ADC performs the conversion by integrating the input signal and correlating the integration time with a digital counter, Known as single- and dual-slope ADCs.
- These types of converters are used in high-resolution applications but have relatively slow conversions.
- However, they are very inexpensive to produce and are commonly found in slow-speed, cost-conscious applications.

Analogue to Digital Converter

.Voltage to Frequency Converter:

$$V_C = \frac{-1}{RC}\int_0^t V_i dt = -\frac{V_i t}{RC}$$

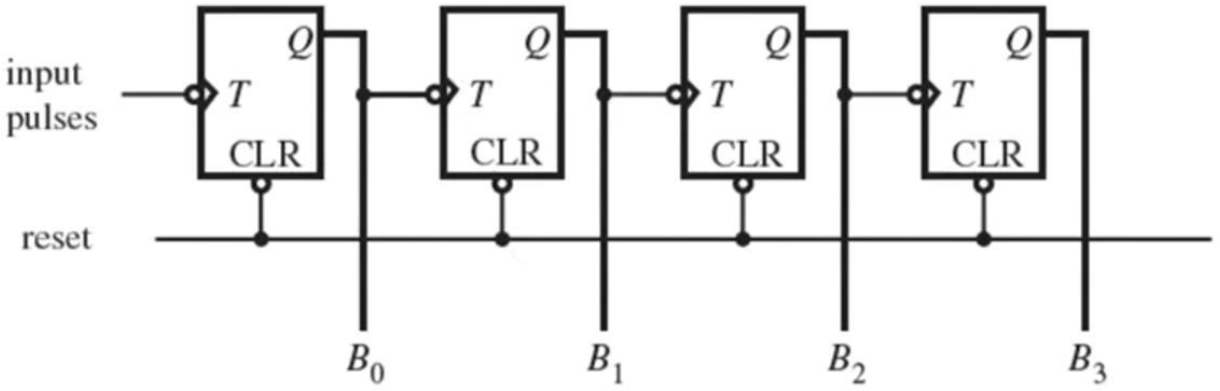


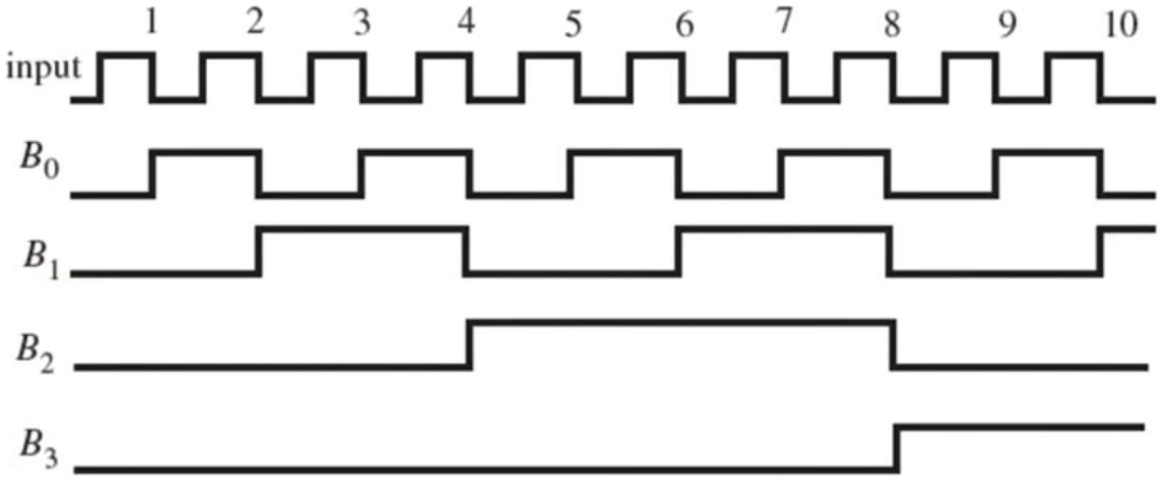
- These voltage-to-frequency converters or voltage controlled oscillators are relatively simple and accurate circuits and have been used for other purposes.
- The voltage across the capacitor is the integral of the current in the noninverting leg of the amplifier.
- This current is proportional to the voltage across R.
- As the voltage on the capacitor rises, a threshold circuit checks this voltage.

Analogue to Digital Converter

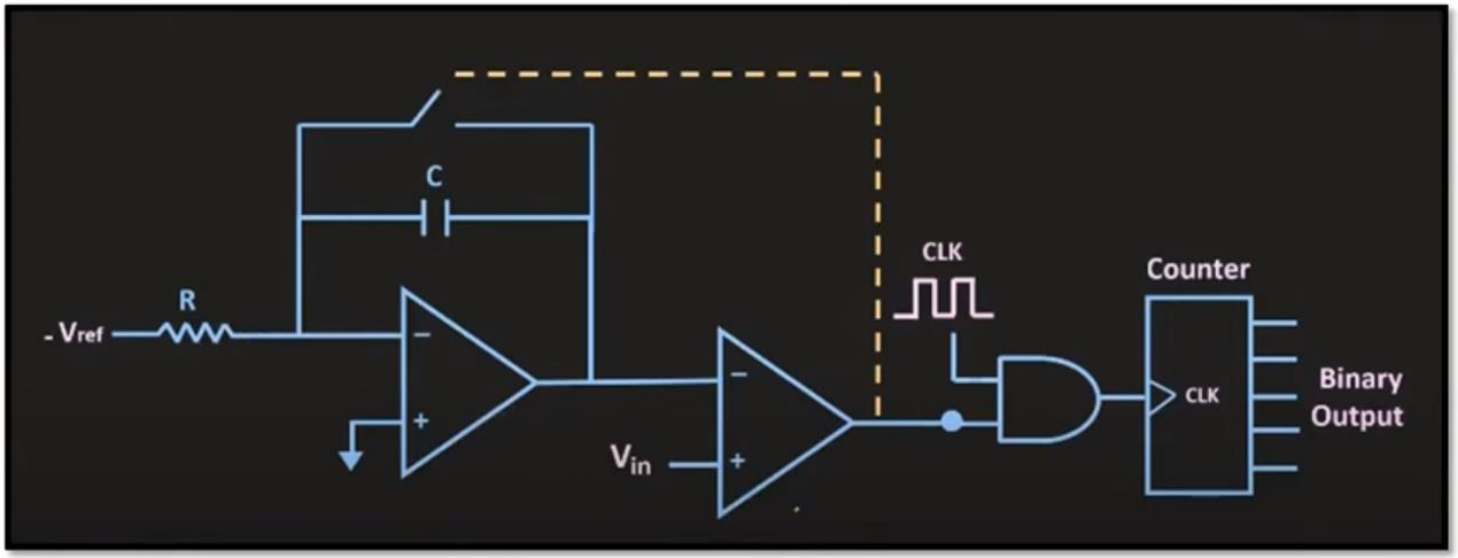
Digital Counter

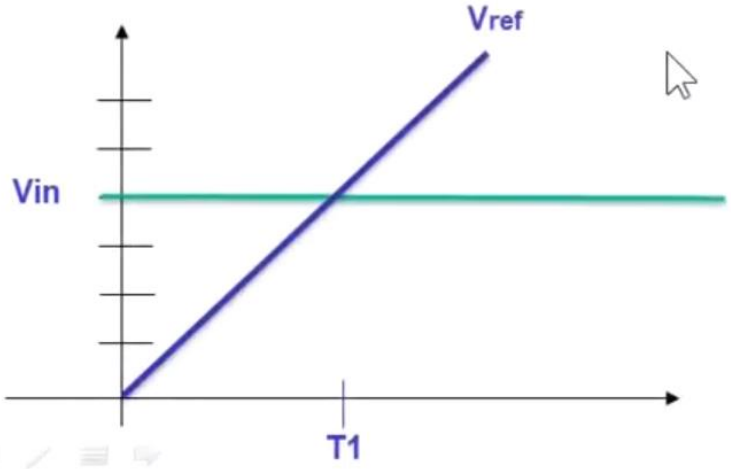
Jk Flip Flop





Single-Slope ADC





$$Vin \dot{\alpha}\ T$$

$$T = \frac{Vin}{LSB} \text{ Tclk}$$

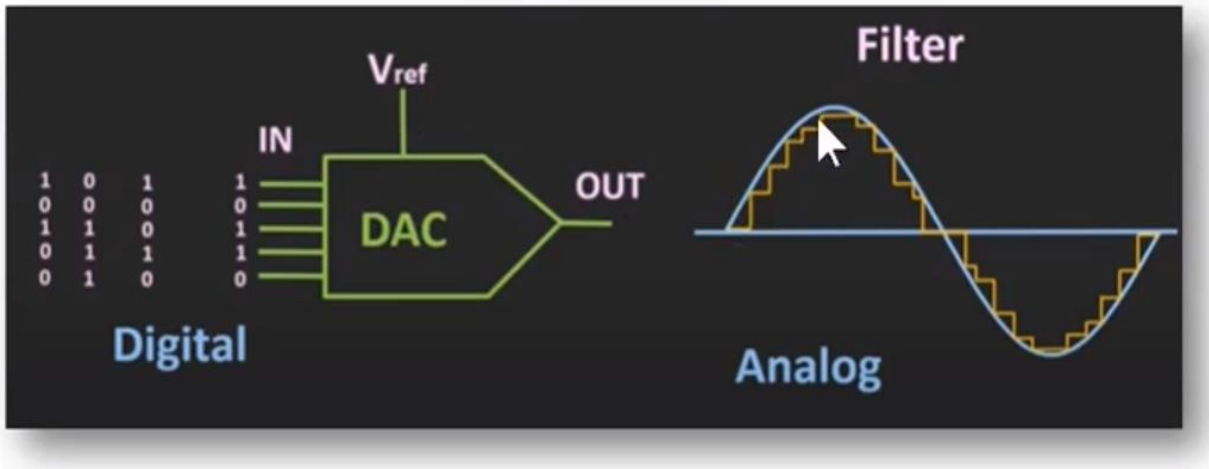
$$\text{LSB} = \frac{VREF}{2^N}$$

$$T = \frac{Vin}{VERF} \text{ Tclk } 2^N$$

Digital to Analogue Converter

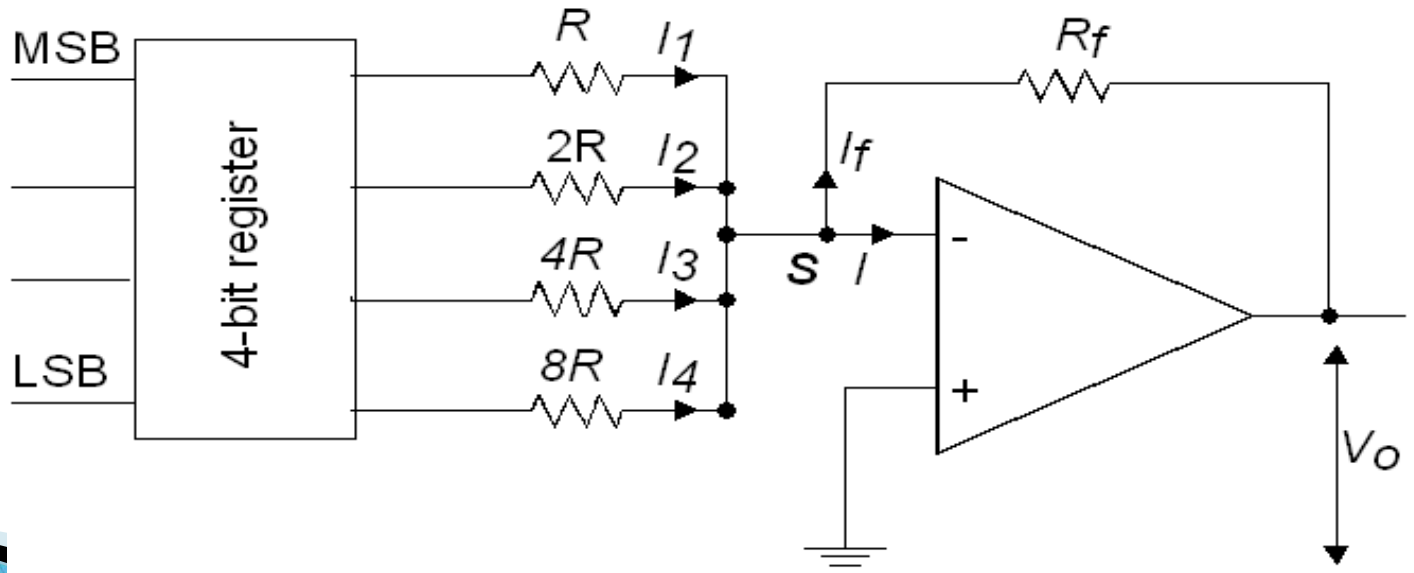
- **D/A conversion** is an important interface process for converting digital signals to analog
- (linear) signals. An example is a voice signal that is digitized for storage, processing, or transmission and must be changed back into an approximation of the original audio signal in order to drive a speaker.

converts a digital number into a corresponding analog voltage or current .



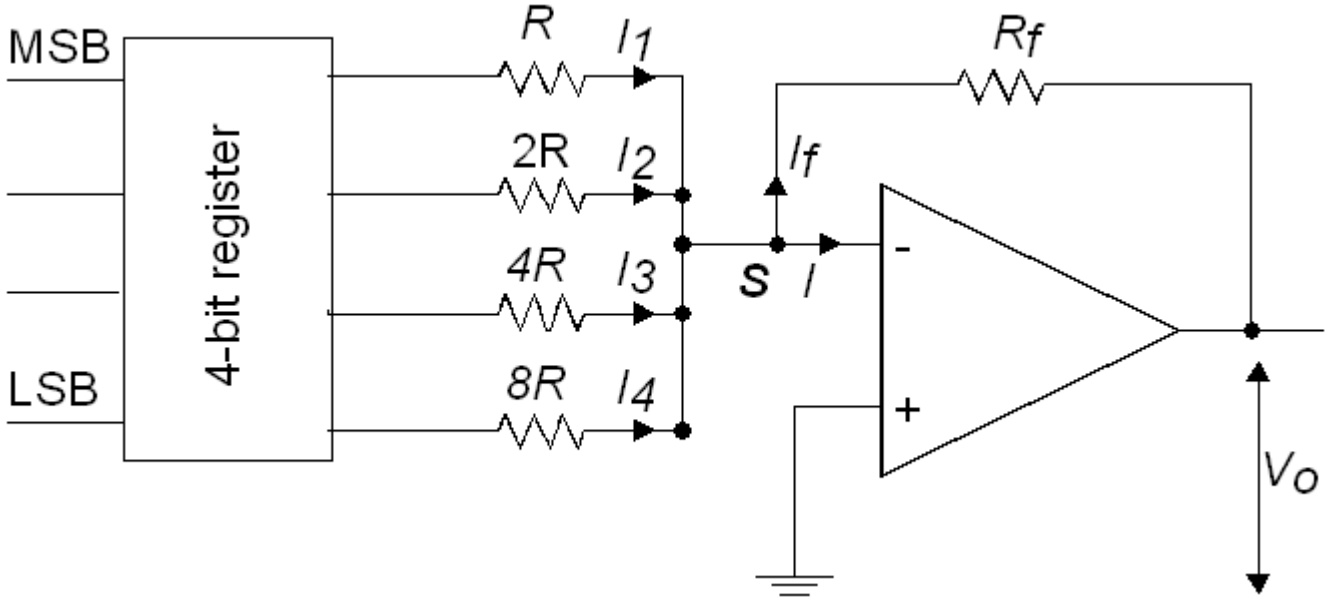
Digital to Analogue Converter

1. Binary Weighted Registers D/A:
   - Comprises of a register and resistor network
   - Output of each bit of the register will depend on whether a '1' or a '0' is stored in that position
   - for a '0' then 0V output
   - for a '1' then 5V output
   - Resistance R is inversely proportional to binary weight of each digit



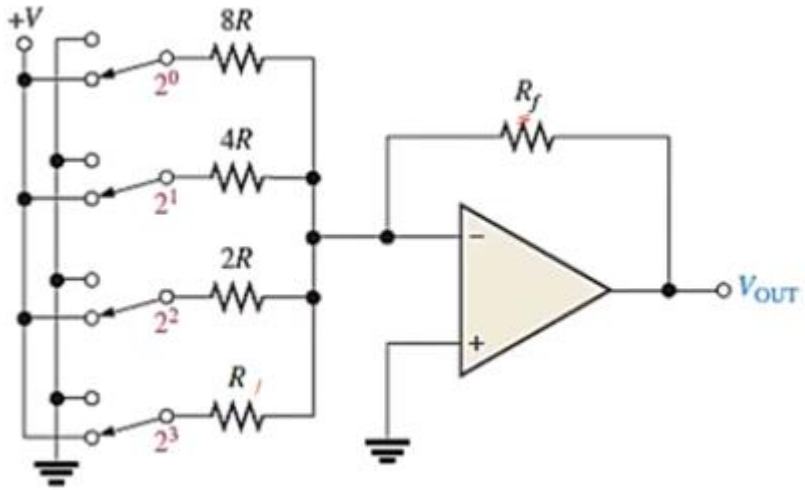
Digital to Analogue Converter

1. Binary Weighted Registers D/A:
   - Best solution is to follow the resistor network with a buffer amplifier
   - Has high impedance, practically no current flows
   - All input currents sum at S and go through R_f
   - $V_o = -I_f R_f$



$$V_o = -I_f \times R_f = -(I_1 + I_2 + I_3 + I_4) \times R_f$$

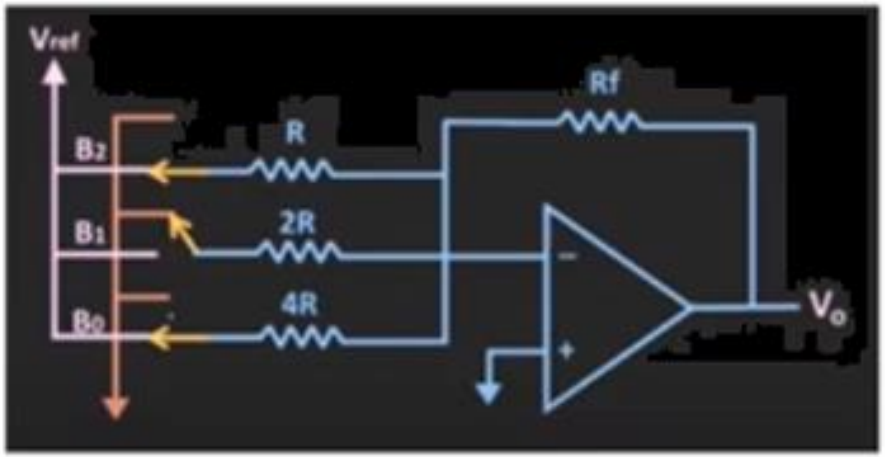
- If the input voltage is zero (binary 0000), the output voltage is also obviously zero. If the input voltage is some other value (from a binary number greater than 0), then the amount of voltage depends on the input resistor value and is different for each resistor.
- The values of the resistors are inversely proportional to the binary weights of the corresponding bits. For example, the lowest value resistor R corresponds to the MSB, 2^3 in this case, which is eight (binary 1000). The other resistors are multiples of R.



$$V_o = -\frac{R_F}{R} V_{REF}(B_3 + \frac{B_2}{2} + \frac{B_1}{4} + \frac{B_0}{8})$$

EXAMPLE

- the circuit diagram of binary weighted DAC with an input voltage =101 ,Calculate the analog output voltage IF Rf = 0.5 R and Vref = 5v .

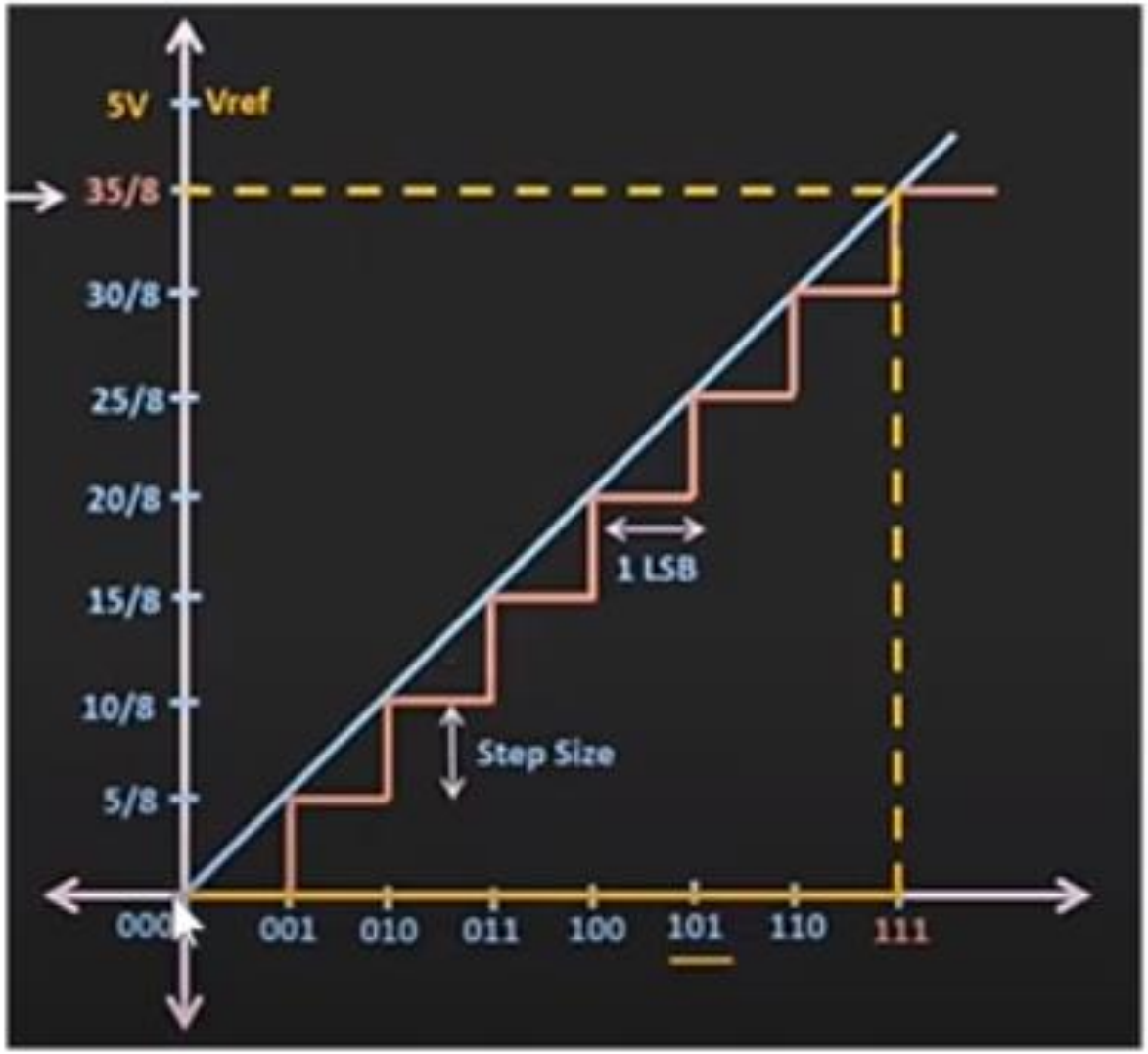


$$V_o = -\frac{R_F}{R} V_{REF}\left(B_3 + \frac{B_2}{2} + \frac{B_1}{4} + \frac{B_0}{8}\right)$$

Example

$$FSO = \frac{(2^n - 1) \times V_{ref}}{2^n}$$

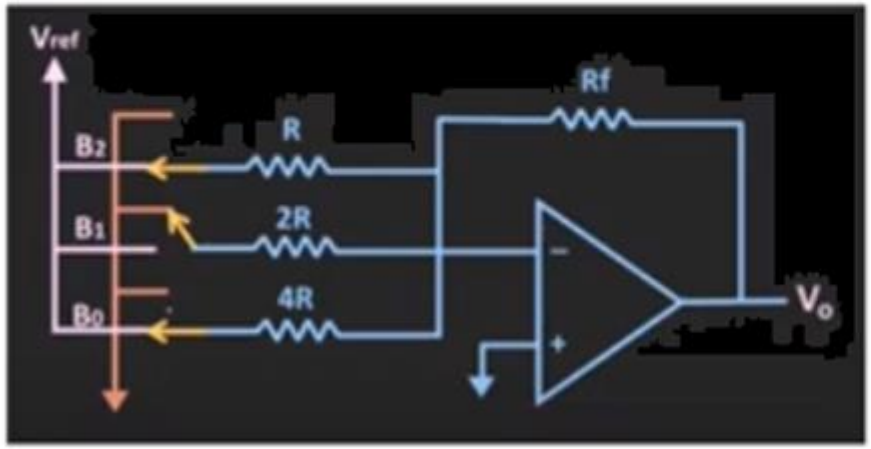
$$FSO = V_{ref} - 1\ LSB$$



Example

- Draw the circuit diagram of a binary weighted DAC with an input voltage = 101, then calculate the analog output voltage. Assume, Rf = R and Vref= **5v**

Solution:



$$V_o = -\frac{R_F}{R} V_{REF}(B_3 + \frac{B_2}{2} + \frac{B_1}{4} + \frac{B_0}{8})$$

Vo= 25/4 = 6.25

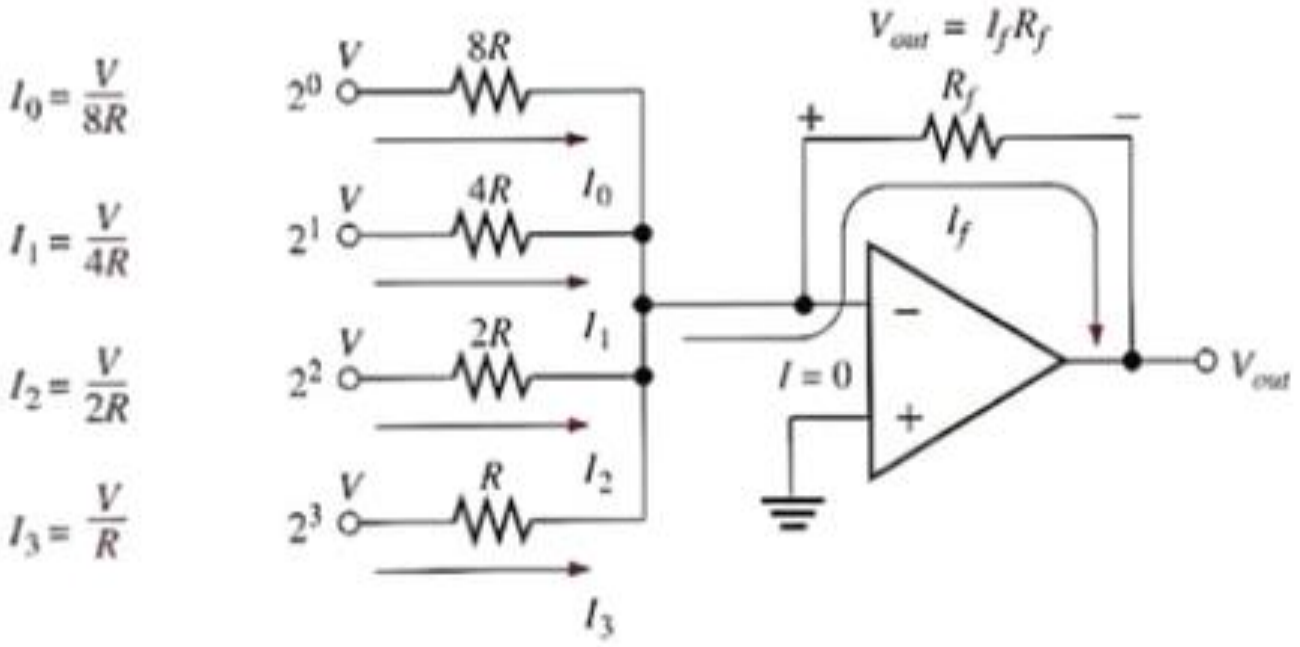
LSB = 5/8 = 0.625

Example 1

For 4-bits binary weighted resistor DAC with given condition R = 10 kΩ, Rf = 5 kΩ, **Vref** = -10 V . Find the following:

1. Output voltage **Vout** for applied binary word is (1001).
2. LSB
3. Full scale output voltage
4. Weight of bits 0 and 3.

$$V_o = -\frac{R_F}{R} V_{REF}(B_3 + \frac{B_2}{2} + \frac{B_1}{4} + \frac{B_0}{8})$$

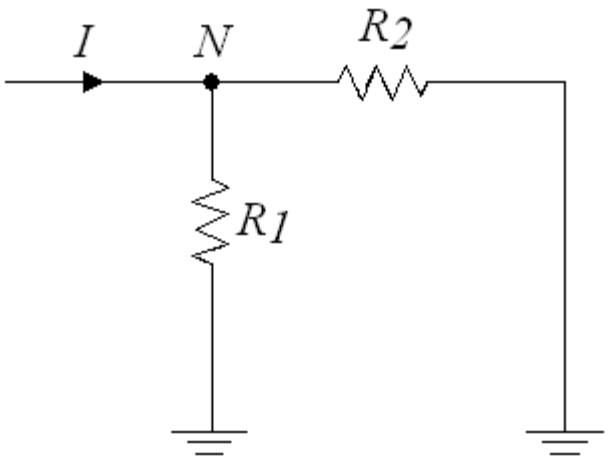


Digital to Analogue Converter

- **disadvantage binary Weighted Registers D/A:**
- Rarely used when more than 6 bits in the code word
- To illustrate the problem consider the design of an 8-bit DAC if the smallest resistor has resistance R
  - what would be the value of the largest resistor?
  - what would be the tolerance of the smallest resistor?
- Very difficult to manufacture very accurate resistors over this range

Digital to Analogue Converter

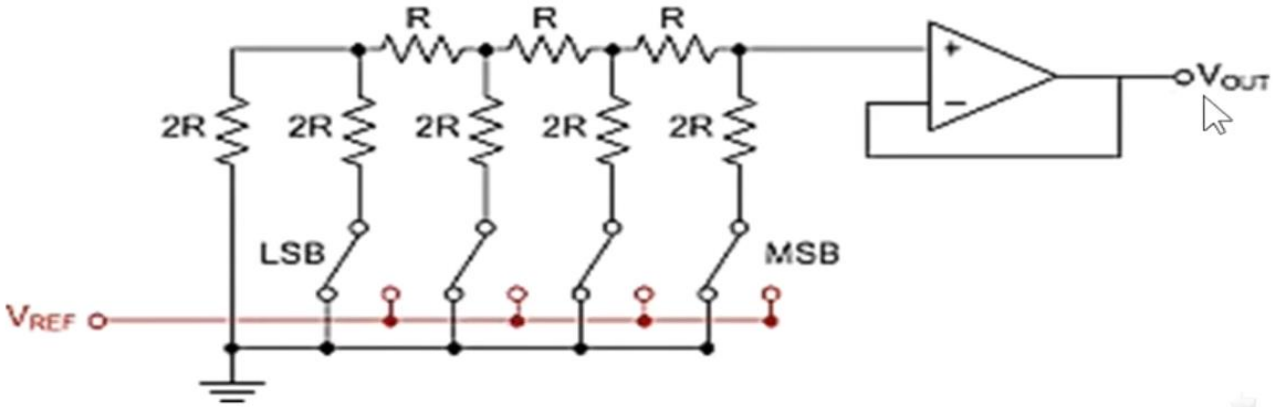
2. The R-2R Ladder:
   - Has a resistor network which requires resistance values that differ 2:1 for any sized code word
   - The principle of the network is based on Kirchhoff's current rule
   - The current entering N must leave by way of the two resistors R1 and R2



Digital to Analogue Converter

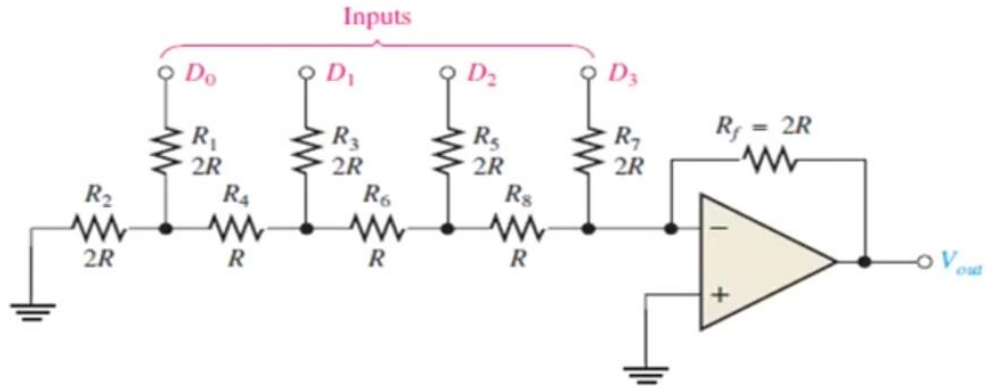
R/2R Ladder DAC

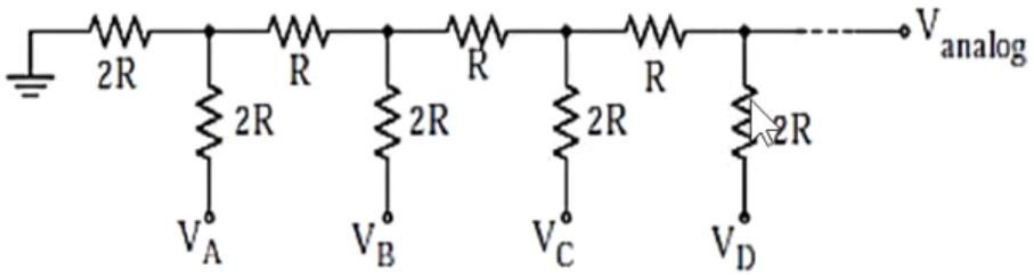
- The R/2R ladder DAC overcomes the problems inherent with the weighted resistor DAC in that it requires only 2 resistor values. These values are standard values that are in production and usually range from 2kΩ to 10kΩ.
- But it requires more resistors and more conversion time than the previous circuit.
- The circuit is expandable to any number of bits simply by adding one resistor of each value for each bit. This is another advantage of the R/2R DAC.



Digital to Analogue Converter

- The output impedance is always equal to R

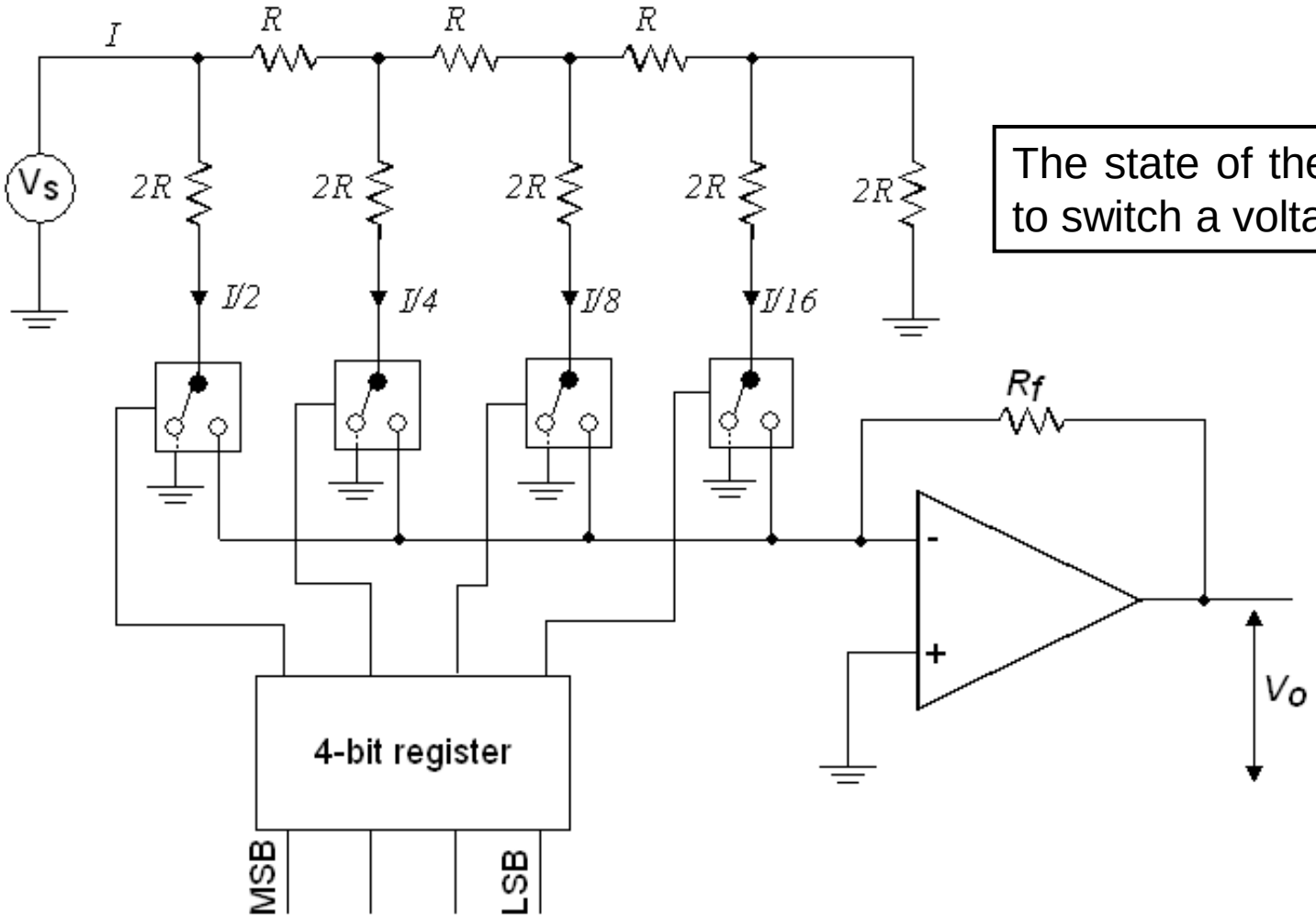




$$V_{analog} = \frac{V_A + 2V_B + 4V_C + 8V_D + \ldots}{2^n}$$

Digital to Analogue Converter

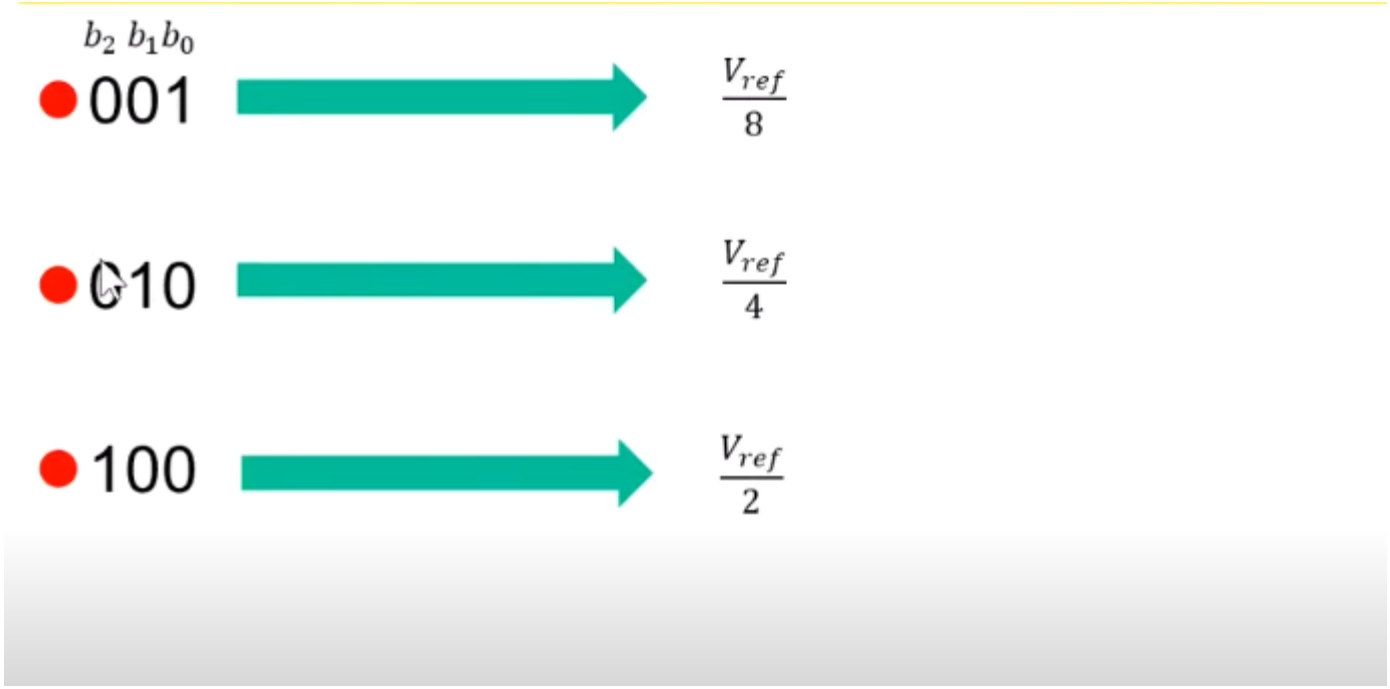
2. The R-2R Ladder:



The state of the bits is used to switch a voltage source

Digital to Analogue Converter

2. The R-2R Ladder:



Real time system
Lecture 7: signal condition

ahmed Mohammed basheer

Systems & Control Engineering Department,
College of Electronics Engineering, Ninevah University.

The Scheduler

The scheduler is at the heart of every kernel. A scheduler provides the algorithms needed to determine which task executes when. To understand how scheduling works, this section describes the following topics:

1)Schedulable entities,

2)Multitasking,

3)Context switching,

4)Dispatcher

5)Scheduling algorithms

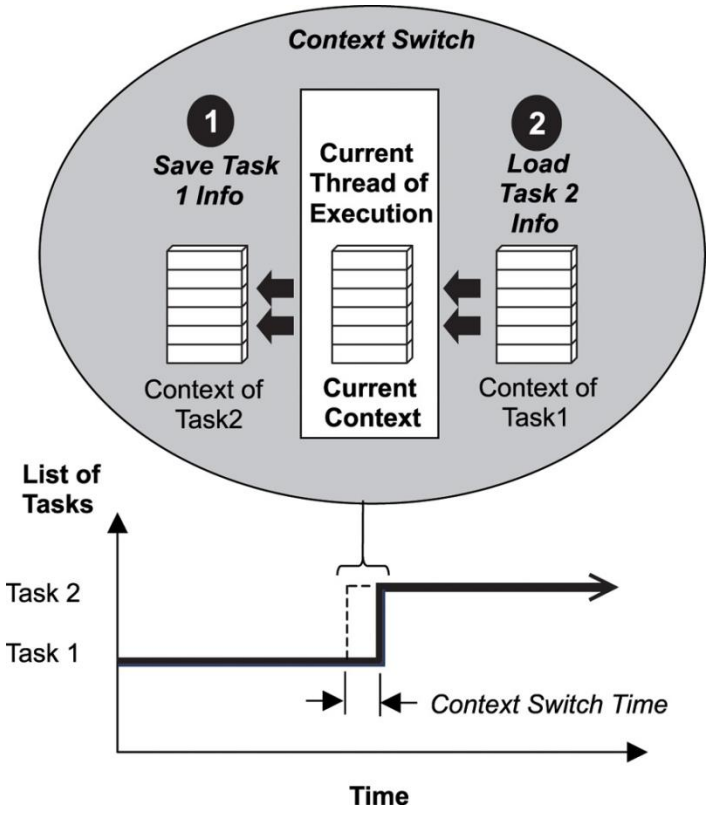
•**Schedulable Entities:** A schedulable entity is a kernel object that can compete for execution time on a system, based on a predefined scheduling algorithm. **Tasks and processes are all examples of schedulable entities found in most kernels.**

A task is an independent thread of execution that contains a sequence of independently schedulable instructions. Some kernels provide another type of a schedulable object called a process. Processes are similar to tasks in that they can independently compete for CPU execution time. Processes differ from tasks in that they provide better memory protection features, at the expense of performance and memory overhead. Despite these differences, for the sake of simplicity, this book uses task to mean either a task or a process.

The Scheduler

•**Multitasking:** Multitasking is the ability of the operating system to handle multiple activities within set deadlines. A real-time kernel might have multiple tasks that it has to schedule to run. One such multitasking scenario is illustrated in Figure. In this scenario, the kernel multitasks in such a way that many threads of execution appear to be running concurrently; however, the kernel is actually interleaving executions sequentially, based on a preset scheduling algorithm. The scheduler must ensure that the appropriate task runs at the right time.

An important point to note here is that the tasks follow the kernel's scheduling algorithm, while interrupt service routines (ISR) are triggered to run because of hardware interrupts and their established priorities.



The Scheduler

•**The Context Switch:** Each task has its own context, which is the state of the CPU registers required each time it is scheduled to run. A context switch occurs when the scheduler switches from one task to another. To better understand what happens during a context switch, let's examine further what a typical kernel does in this scenario.

Every time a new task is created, the kernel also creates and maintains an associated task control block (TCB). TCBs are system data structures that the kernel uses to maintain task-specific information. **TCBs contain everything a kernel needs to know about a particular task.** When a task is running, its context is highly dynamic. This dynamic context is maintained in the TCB. When the task is not running, its context is frozen within the TCB, to be restored the next time the task runs. A typical context switch scenario is illustrated in the previous Figure

As shown in the previous Figure, when the kernel's scheduler determines that it needs to stop running task 1 and start running task 2, it takes the following steps:

1)The kernel saves task 1's context information in its TCB. 2)It loads task 2's context information from its TCB, which becomes the current thread of execution.

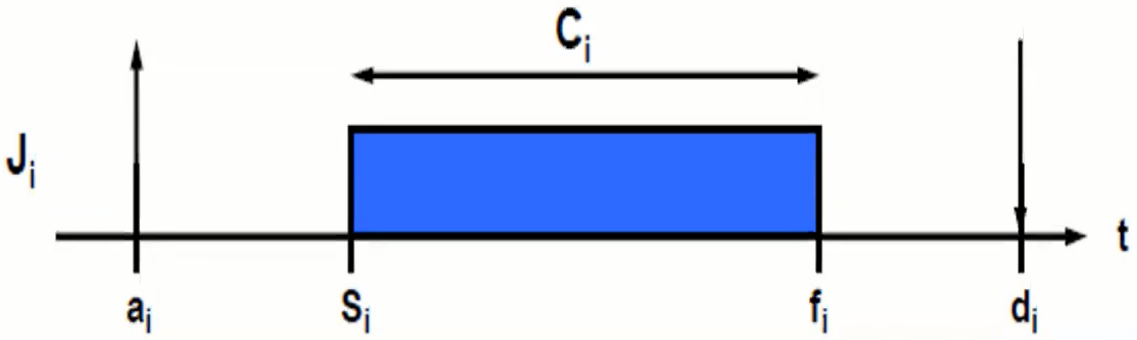
3)The context of task 1 is frozen while task 2 executes, but if the scheduler needs to run task 1 again, task 1 continues from where it left off just before the context switch.

The Scheduler

•**The Dispatcher:** The dispatcher is the part of the scheduler that performs context switching and changes the flow of execution. At any time an RTOS is running, the flow of execution, also known as flow of control, is passing through one of three areas: through an application task, through an ISR, or through the kernel. When a task or ISR makes a system call, the flow of control passes to the kernel to execute one of the system routines provided by the kernel. When it is time to leave the kernel, the dispatcher is responsible for passing control to one of the tasks in the user's application. It will not necessarily be the same task that made the system call. It is the scheduling algorithms (to be discussed later) of the scheduler that determines which task executes next. It is the dispatcher that does the actual work of context switching and passing execution control.

Scheduler: Choose the next process to run

ai: Arrival time
ci: Computation time
di: Deadline
Si: Start time
fi: Finishing time



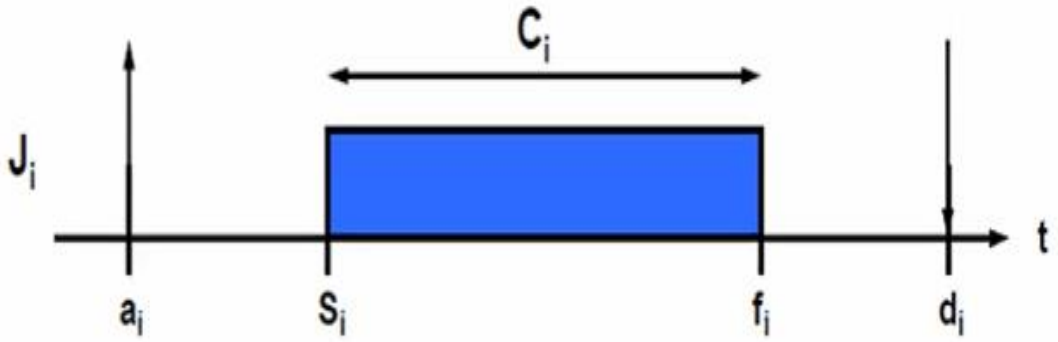
Typical parameters of a real-time task

Scheduler: Choose the next process to run

ai: Arrival time
ci: Computation time
di: Deadline
Si: Start time
fi: Finishing time

You are going to bring bread from a bakery shop. You arrived at 10:00 and the bakery shop started serving you at 10:15 and finished at 10:20 and you need to leav by 10:30. Which one is true

- Arrival time=10:00, burst time=5 minutes, finish time=10:30, deadline=10:20
- Arrival time=10:00, computation time=15 minutes, finish time=10:20, deadline=10:30
- Arrival time=10:00, burst time=5 minutes, finish time=10:20, deadline=10:30
- Arrival time=10:15, computation time=15 minutes, finish time=10:20, deadline=10:30



Typical parameters of a real-time task

CPU Scheduling

1st Case : FCFS (First Come First Served)

Suppose that the processes arrive at time 0,

Draw Gantt Chart and calculate the average waiting time using the given table ??

Process	Burst Time
P1	3
P2	5
P3	9
P4	7

CPU Scheduling

1st Case : FCFS (First Come First Served)

Suppose that the processes arrive at time 0,

Draw Gantt Chart and calculate the average waiting time using the given table ??

Process	Burst Time
P1	3
P2	5
P3	9
P4	7

Waiting time :

P1 = 0
P2 = 3
P3 = 8
P4 = 17

Average waiting time = (0 + 8 + 3 + 17) / 4 = 7

CPU Scheduling

1st Case : FCFS (First Come First Served)

Suppose that the processes arrive at time 0, in the order: P1 , P3 , P2 , P4

Draw Gantt Chart and calculate the average waiting time using the given table ??

Process	Burst Time
P1	3
P2	9
P3	5
P4	7

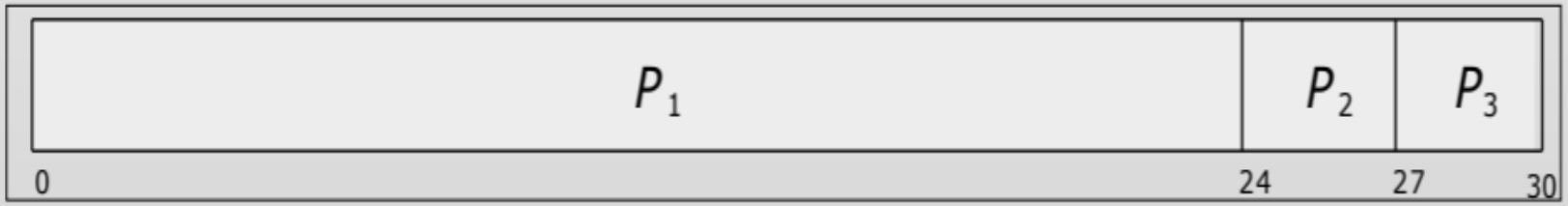
Waiting time :

P1 = 0
P2 = 8
P3 = 3
P4 = 17

Average waiting time = (0 + 8 + 3 + 17) / 4 = 7

Example 1: Suppose that the processes arrive at time 0, in the order: P_1 , P_2 , P_3 Draw the Gantt chart and calculate the average waiting?

Processes	Burst time
P1	24
P2	3
P3	3



- Waiting time:
  P_1 = (0-0)=0
  P_2 = (24-0)=24
  P_3 = (27-0)=27

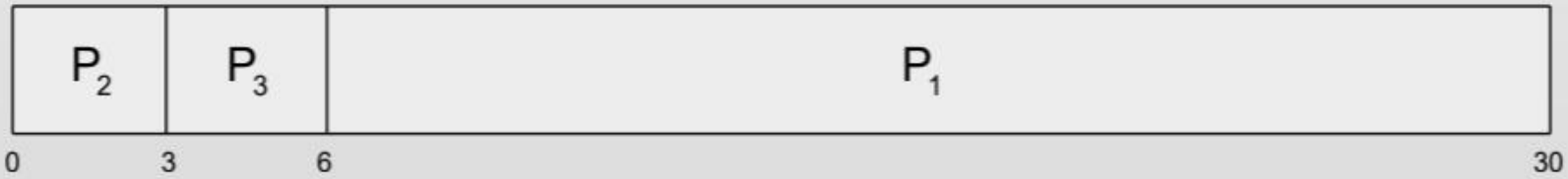
Waiting time= start time – arrival time ▪
وقت الانتظار= وقت بدأ التنفيذ – وقت الوصول

- Average waiting time: (0 + 24 + 27)/3 = 17 Ms.

FCFS Scheduling (Cont.)

Example 2: Suppose that the processes arrive at time 0, in the order: P_2 , P_3 , P_1 Draw the Gantt chart and calculate the average waiting?

Processes	Burst time
P1	24
P2	3
P3	3



- Waiting time:

  $P_1 = 6$
  $P_2 = 0$
  $P_3 = 3$

- Average waiting time: (6 + 0 + 3)/3 = 3 Ms.

Much better than previous case

2nd Case : FCFS (First Come First Served)

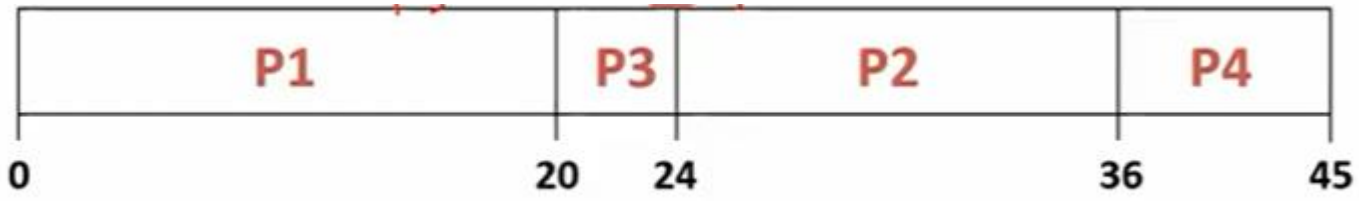
Draw Gantt Chart and calculate the average waiting time using the given table ??

Process	Burst Time	Arrival Time
P1	20	0
P2	12	3
P3	4	2
P4	9	5

Waiting time : start time – arrival time

P1 = 0 – 0 = 0
P2 = 24 – 3 = 21
P3 = 20 – 2 = 18
P4 = 36 – 5 = 31

Average waiting time = (0 + 21 + 18 + 31) / 4 = 70 / 4



3rd Case : SJF (short job first) non-Preemptive

Draw Gantt Chart and calculate the average waiting time using the given table ??

Process	Burst Time	Arrival Time
P2	12	0
P3	8	3
P4	4	5
P1	10	10
P5	6	12

Waiting time : start time – arrival time

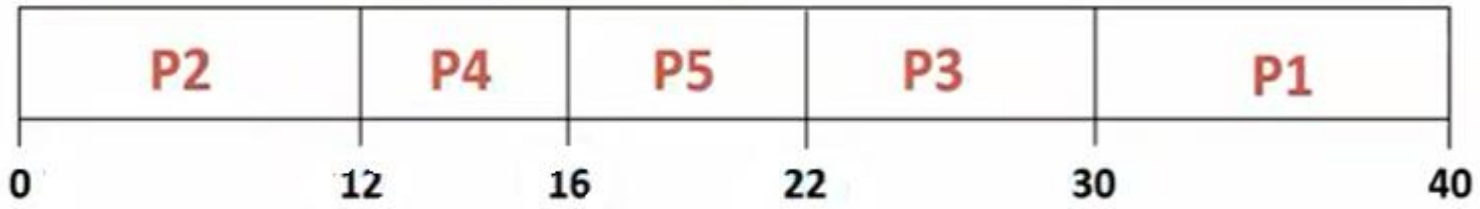
P1 = 30 – 10 = 20

P2 = 0 – 0 = 0

P3 = 22 – 3 = 19

P4 = 12 – 5 = 7

P5 = 16 – 12 = 4



Average waiting time = (20 + 0 + 19 + 7 + 4) / 5 = 50 / 5 = 10

Non-preemptive scheduling

- Each process completes its full CPU burst cycle before the next process is scheduled.
- No time slicing or CPU stealing occurs.
- Once a process has control of the CPU, it keeps control until it gives it up (e.g. to wait for I/O or to terminate).
- Works OK for batch processing systems, but not suitable for time sharing systems.

Preemptive scheduling

- A process may be interrupted during a CPU burst and another process scheduled.
- More expensive implementation due to process switching.
- Used in all time sharing and real time systems.

4th Case : SJF (short job first) Preemptive

Draw Gantt Chart and calculate the average waiting time using the given table ??

Process	Burst Time	Arrival Time
P2	12	0
P3	8	3
P4	4	5
P1	10	10
P5	6	12

Process	Burst Time	Arrival Time
P2	12 9	0
P3	8 6	3
P4	4	5
P1	10	10
P5	6	12

Waiting time : start time – arrival time

P1 = 30 – 10 = 20

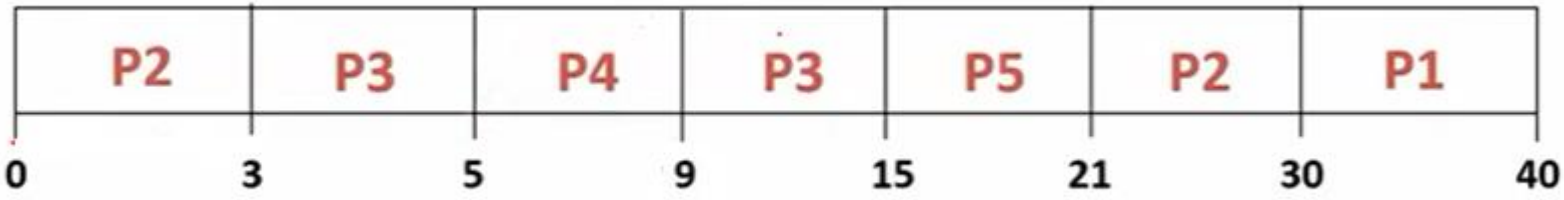
P2 = (0 – 0) + (21 - 3) = 18

P3 = (3 – 3) + (9 - 5) = 4

P4 = (5 – 5) = 0

P5 = 15 – 12 = 3

Average waiting time = (20 + 18 + 4 + 0 + 3) / 5 = 45 / 5 = 9



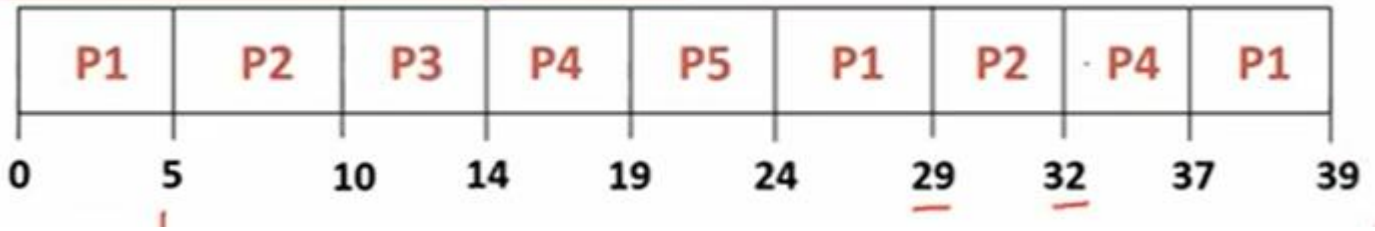
5th Case : Round Robin (RR)

Draw Gantt Chart and Calculate The Average Waiting Time , where **Quantum = 5 ms**

Process	Burst Time
P1	12
P2	8
P3	4
P4	10
P5	5

Waiting time :

P1 = 0 + (24 - 5) + (37 - 29) = 27
P2 = 5 + (29 - 10) = 24
P3 = 10
P4 = 14 + (32 - 19) = 27
P5 = 19



Average waiting time = (27 + 24 + 10 + 27 + 19) / 5 = 107 / 5 = 21.4

Main purposes of scheduling

- Maximize CPU Utilization: Keep the CPU as busy as possible
- Minimize resource starvation: No resource takes full utilization of CPU
- Ensure fairness: Each process gets fair share of the CPU time.
- Minimize waiting time

Home work

Draw the Gantt Chart for the following task given in the table using:

- Short Job First Non-preemptive
- Short Job First Preemptive
- Round robin (quantum =4)

Process	Computation Time	Arrival Time
P1	15	0
P2	12	3
P3	4	2