

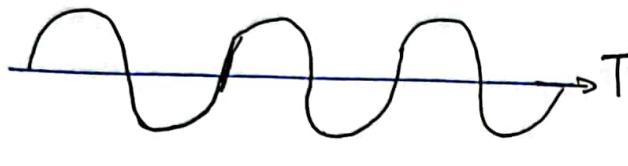
Analog-To-Digital-Converter (ADC)

77

* Signal types:

1-analog signal

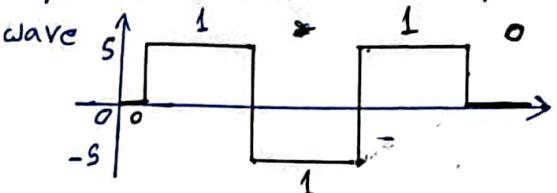
- physical quantity varies with time in a smooth and continuous fashion
- Value is continuously changing



2-digital signal

- Binary digital signals use two discrete voltage levels to represent binary 1 or 0

→ sequence of fixed-width square



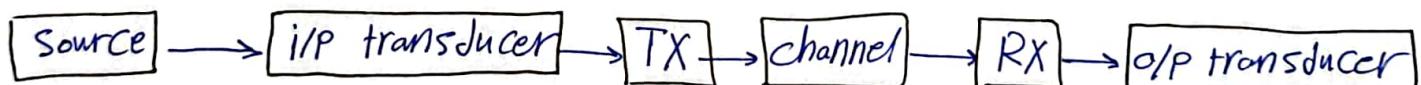
ADC: is a circuit that converts a continuous voltage value (analog) to a binary value (digital) that can be understood by a digital device which could then be used for digital computation.

→ these ADC circuits can be found as individual ADC ICs by themselves or embedded into a microcontroller

* digital to analog converter (DAC): - are used in transforming transmitted or stored data or the results of digital processing, back to "real world"

Ex:-

elements of communication system :-



I/P transducer: converts the message produced by a source to a form suitable for the communication system (analog to digital)

O/P transducer: converts the received signal back into a useful quantity
OR: converts digital electrical signal into the form desired by the system (analog)

Ex:-

I/P transducer : speech waves → microphone → Voltage
(Modulation process)

O/P transducer : Voltage → Loudspeakers (Demodulation process)

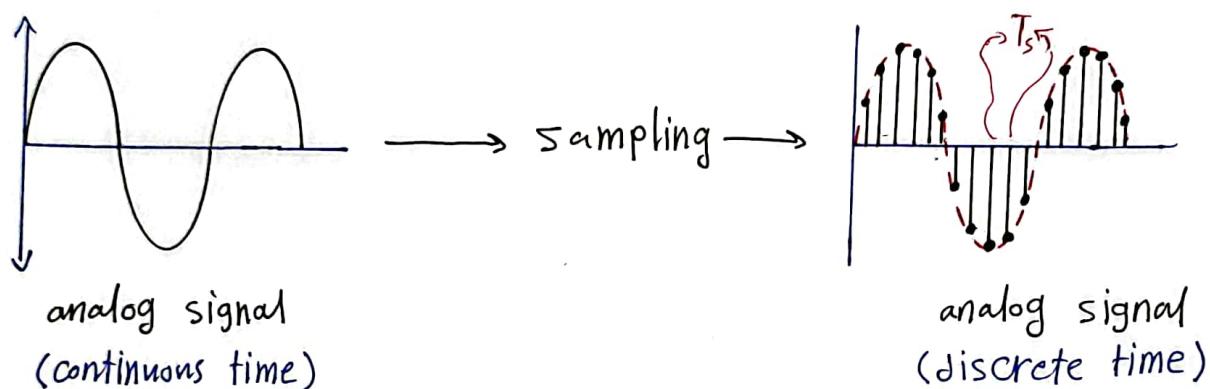
- * In Converting an analog signal to an equivalent sequence of "0" and "1" (digital signal) we go through three processes:
 - 1- Sampling
 - 2- Quantization
 - 3- Coding (symbol to bit mapper, pulse code modulation)

→ these three processes are the part of modulation process. (PCM)

→ Modulation:-
the process of converting the message (signal) to a form suitable for the system

1- Sampling :-

→ converting continuous-time analog signal to discrete-time analog signal



* the more often a signal is sampled, the better the digital representation of the analog signal

→ Sampling rate: $\overset{F_s}{\curvearrowright}$ refers to the number of samples per second taken from a continuous signal to convert it into a digital signal, the higher the sampling rate, the more accurate the digital representation of the analog signal. this sampling rate must be (at least) twice the highest frequency to accurately represent the signal. $F_s > 2 F_{max}$

→ Nyquist rate: - (F_N) → this is the minimum sampling rate required to accurately represent a signal without aliasing. it is equal to twice the maximum frequency of the signal so :

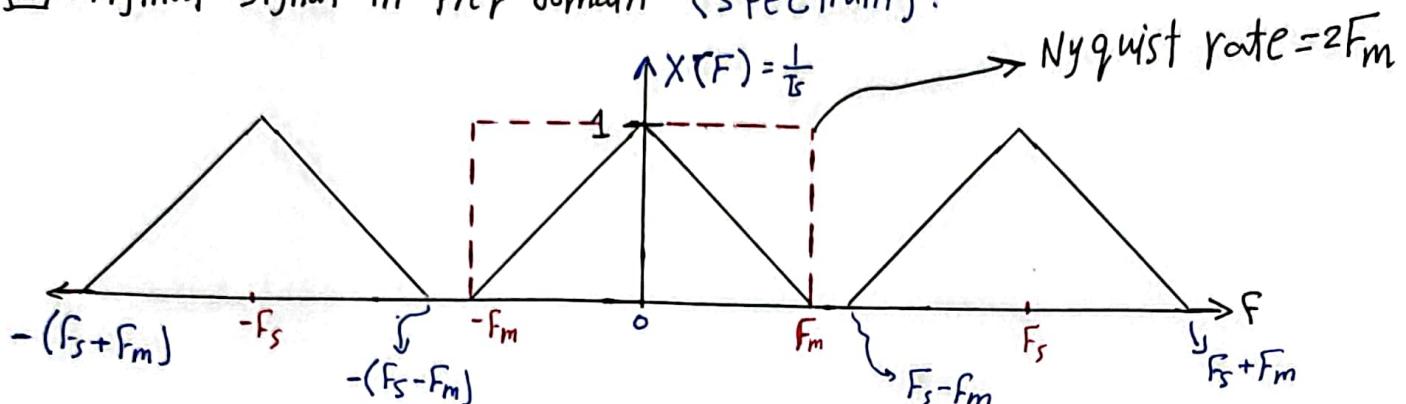
$$\rightarrow F_s \geq 2 F_{max} \quad F_N = 2 F_{max} \quad \therefore F_s > F_N$$

$$F_s = \frac{1}{T_s} \quad \text{sample period}$$

* aliasing problem

→ the signal in frequency domain ~~at~~ not time domain :

① original signal in freq domain (spectrum) :



if →

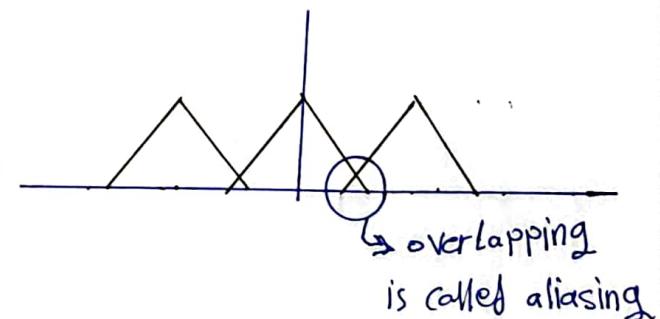
$$\rightarrow F_s - F_m = F_m \rightarrow F_s = 2F_m = F_N \rightarrow \text{minimum requirement for reconstruction of signal}$$

if →

$$F_s - F_m < F_m \rightarrow F_s < 2F_m \rightarrow \text{reconstruction not possible because of aliasing problem}$$

→ that is, the sampling frequency must be more than twice the value of the highest frequency

→ aliasing can cause distortion of the signal



② Quantization :

→ converting discrete-time analog signal to discrete time digital signal

* step size of a quantizer is : the size of the smallest change or increment between adjacent output values.

$$\text{step size } (\Delta) = \frac{V_{\max} - V_{\min}}{L}$$

$$L : \text{No. of quantization level} = 2^m$$

$$m : \text{No. of bits in ADC} \rightarrow \text{(resolution)}$$

ex:-

assume that all values of the analog input fall within a range of 0-5V and we have a 10-bit A/D system

So:-

- * No. of quantization levels "L" = 2^{10} = 1024 levels
- * step size " Δ " = $\frac{5-0}{1024} = 4.8828 \text{ mV}$

CX: 3-bit system :-

$$L = 2^3 = 8 \text{ level}$$

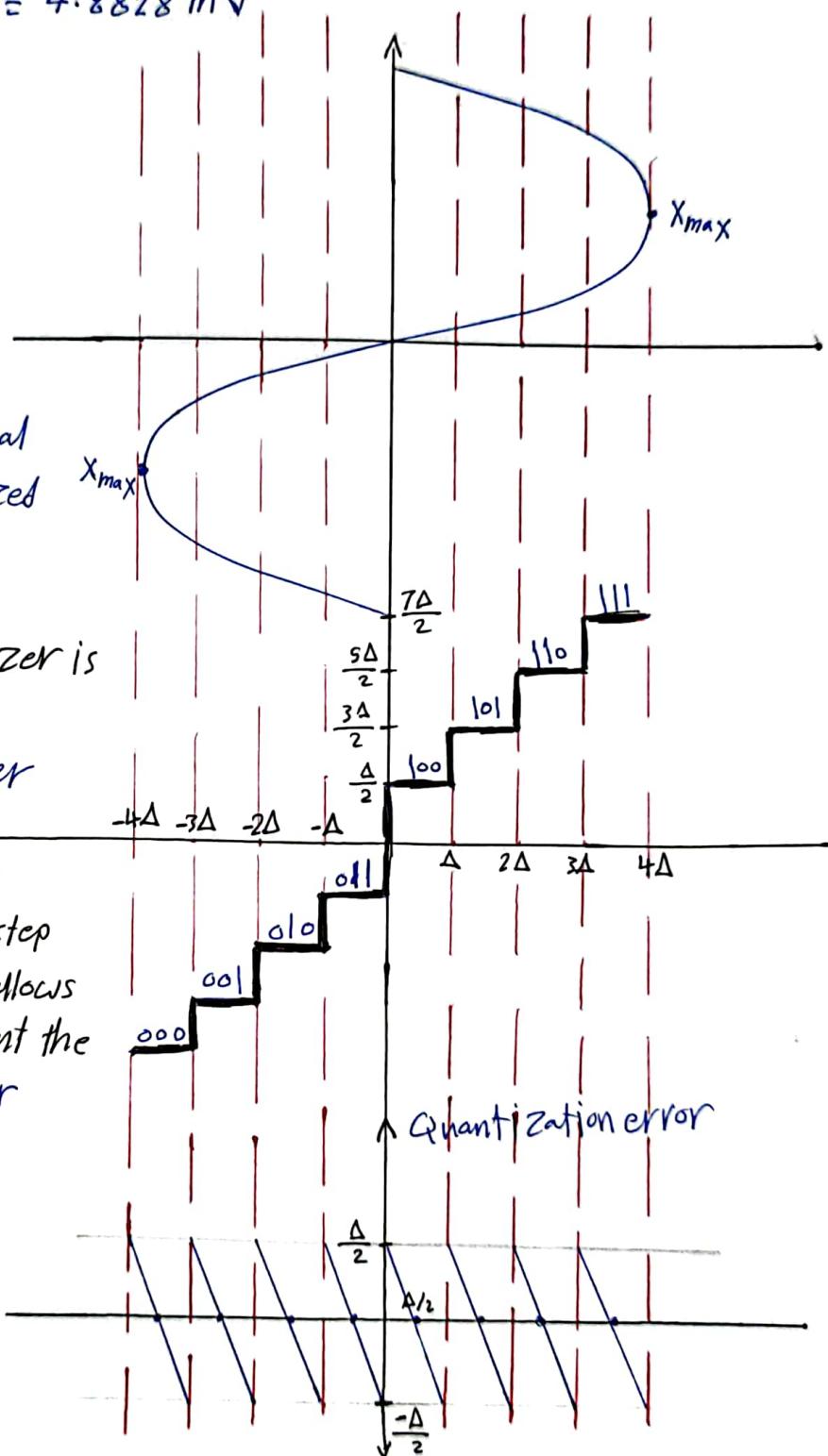
$$\Delta = \frac{2x_{\max}}{2^8}$$

Quantization error:- the

- * difference between the actual input signal and the quantized output signal

* the step size of the quantizer is directly proportional to the Quantization error, A smaller step size leads to a smaller quantization error

→ this is because a smaller step size ~~less~~ allows the quantizer to represent the input signal with greater precision and accuracy



* Parallel VS Serial ADCs

1- Parallel ADCs:- or (flash ADCs)

→ use a set of parallel data lines to transmit the digital output of the ADC

→ In a parallel ADC, each bit of the digital output is transmitted on a separate data line, resulting in a fast conversion time

→ Parallel ADCs have a faster conversion time and are better suited for applications where high-speed data is required.

→ So they require more pins to interface with MCU or other digital device.

→ in case of 16-bit parallel ADC chip, we need 16 pins for the data path

→ in order to save pins, many 12- and 16-bit ADCs use pins D0:D7 to send out the upper and lower bytes of the binary data

→ Parallel ADCs usually offer higher accuracy, because they can sample and hold the input signal over longer periods of time, allowing for better resolution and ~~precision~~ precision.

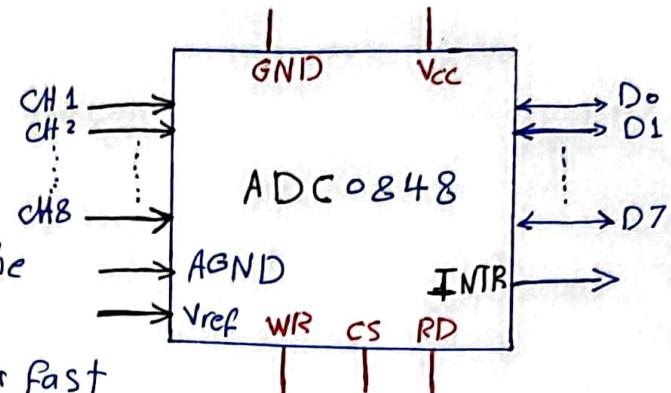
→ the ability to sample and hold an input signal for longer periods of time. Can be important in applications where the signal changes slowly over time or where it is necessary to capture specific features in the signal. For example, in scientific experiments or medical monitoring systems, it may be necessary to sample and hold the input signal over several minutes or hours to capture important information or detect patterns.

→ parallel ADCs are often used in high-speed data systems or other applications where fast conversion rates with high accuracy are required.

* Disadvantages of Parallel ADCs:-

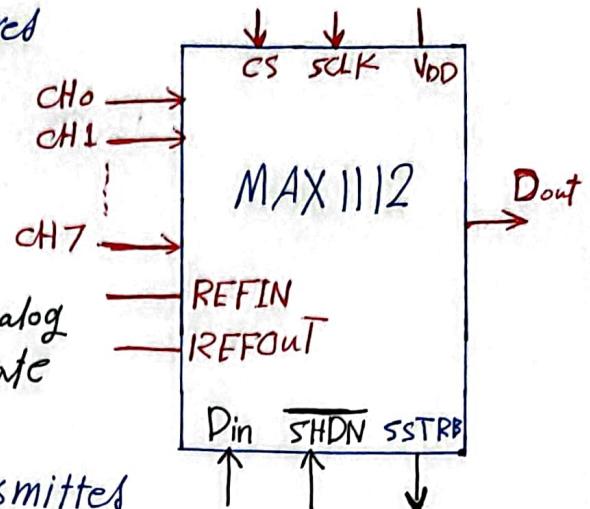
- Complexity - requiring more components
- consuming more power.

- larger in size



② serial ADCs:-

- serial ADCs are simpler in design compared to parallel ones. They convert one bit at a time using a single conversion channel, which makes them easier to implement and less expensive.
- the input signal is sampled using an analog multiplexer that selects the appropriate channel for conversion
- the converted digital data is then transmitted serially through a single output pin.
- that means that inside the serial ADC, there is a parallel-in-serial-out shift register responsible for sending out the binary data one bit at a time.
- So, more CPU time is needed to get the converted data from the ADC because the CPU must get data one bit at a time, instead of in one single read operation as with the parallel ADC.
- One of the main advantages of serial ADCs is their low power consumption because they process only one bit at a time, this makes them ideal for battery-operated devices where power efficiency is critical.
- serial ADCs may have lower accuracy due to their limited sampling rate and resolution when compared with parallel ADCs.
- Serial ADCs require fewer pins and are often smaller and less expensive than parallel ADCs, they are well suited for application where space and cost are important and data speed is not critical



* Resolution :-

- resolution in ADC refers to the number of bits (8, 10, 12, 16, 24) that can be used to represent a digital value for each analog input voltage level.
- the resolution determines how an ADC can divide up the range of possible analog input voltage into discrete steps, which affects its accuracy.
- Higher-resolution ADCs provide a smaller step size (quantization level).
- Step size is the smallest change that can be discerned by an ADC.
- we can control the step size with the help of V_{ref} .
- V_{ref} is the maximum input voltage used for comparing the input voltage.
- the input voltage can be equal or less than V_{ref}

Ex:

If $V_{ref} = 5V$ and

n-bit	No. of steps	Step size
8	$2^8 = 256$	$5/256 = 19.53mV$
10	$2^{10} = 1024$	$5/1024 = 4.88mV$
12	$2^{12} = 4096$	$= 1.2mV$
16	$2^{16} = 65,536$	$= 0.076mV$

Ex for controlling the step size :

In 8-bit ADC $\rightarrow 2^8 = 256$

V_{ref}	Step size
5.00V	$5/256 = 19.53mV$
4.00V	$4/256 = 15.62mV$
3.00V	$3/256 = 11.71mV$
2.56V	$2.56/256 = 10mV$
2.00V	$2/256 = 7.81mV$

→ In 10-bit ADC $\rightarrow 2^{10} = 1024$

5.00V	$5/1024 = 4.88mV$
4.096 5.6 V	$4.096/1024 = 4mV$
3V	$3/1024 = 2.93mV$
2.56V	$2.56/1024 = 2.5mV$

* Conversion Time:-

- is defined as the time it takes the ADC to convert the analog input to a digital number.
- the conversion time is dictated by the clock source connected to the ADC in addition to the method used for data conversion and technology used in the fabrication of the ADC

* digital data output:

- in 8-bit ADC → we have 8-bit digital data → D₀:D₇
- in 10-bit ADC → D₀:D₉

$$D_{out} = \frac{V_{in}}{\text{step size}} = \frac{V_{in}}{(V_{ref}/2^{n-1})}$$

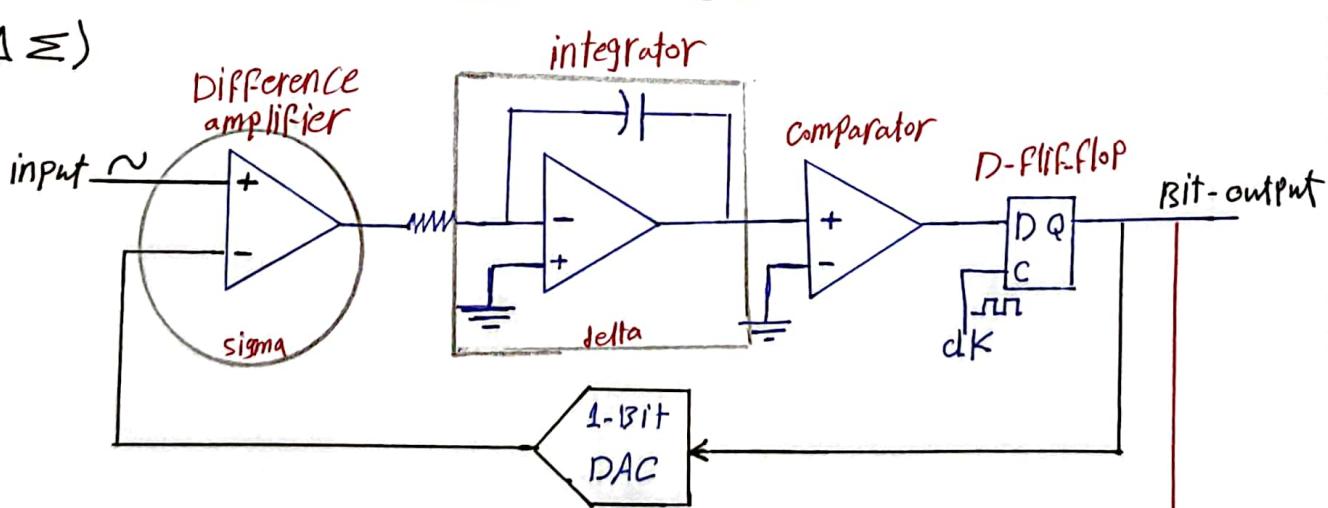
analog input voltage



* ADC Types:-

- parallel design (Flash ADC)
- DAC based design → ramp, SAR, tracking
- integrator based design → single-slope, dual-slope
- sigma-delta design (SD)
- pipelined design.

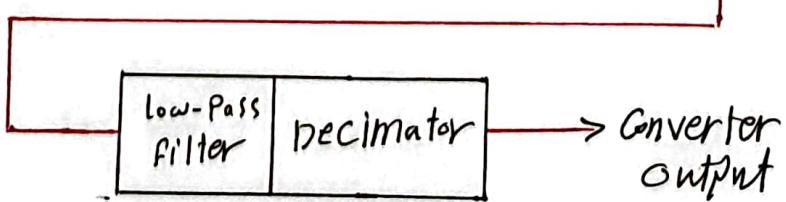
1) ($\Delta \leq$)



→ Delta-Sigma ADC:

low-middle speed (few kSps)

high resolution → ~ 24 bits



* Delta-sigma ($\Delta\sigma$) ADCs are a type of oversampling analog to digital converter that uses a technique called sigma-delta modulation

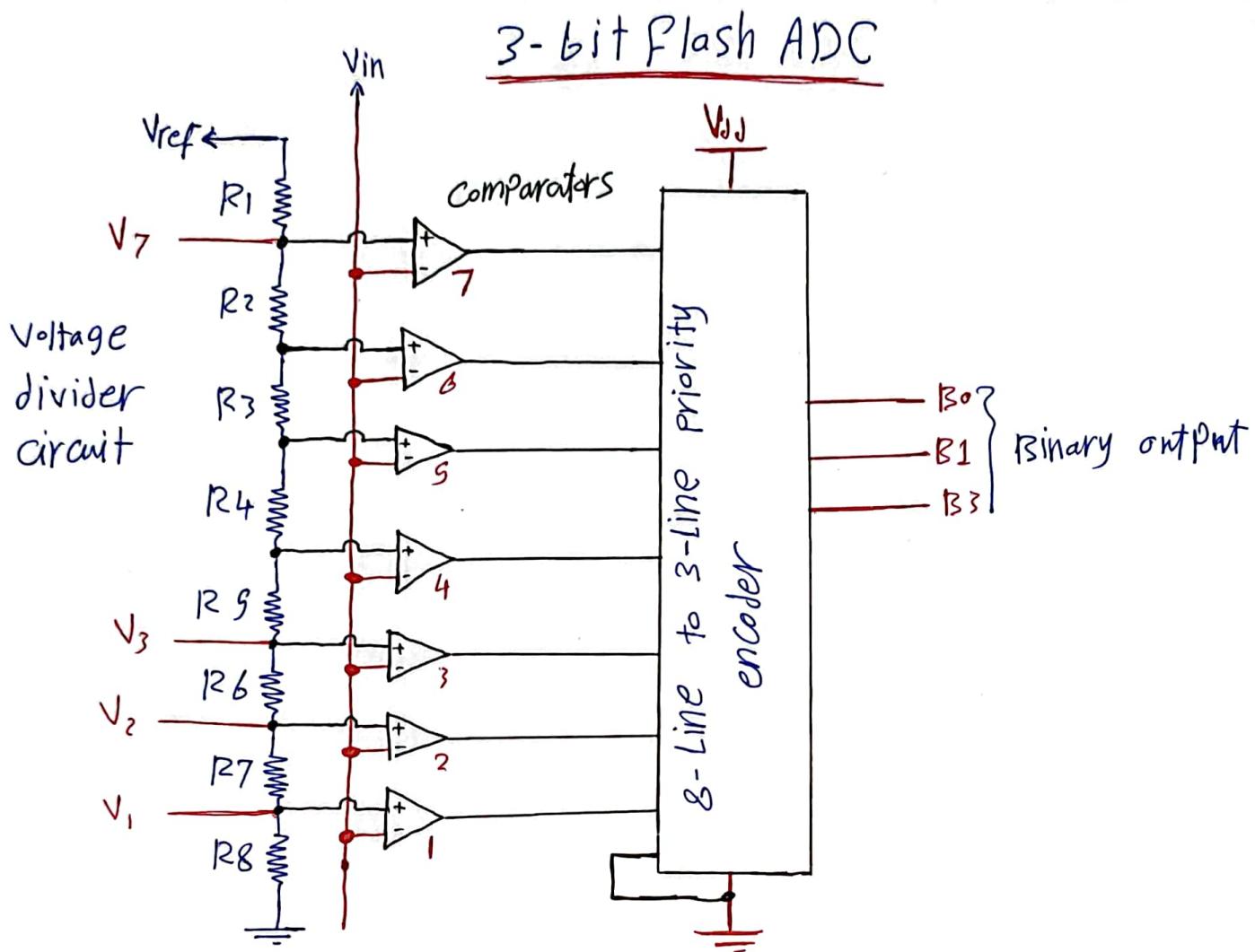
→ the basic principle is to take multiple measurements of the analog signal and then average those measurements over time. this process effectively increases the resolution of the converter by reducing quantization noise with the help of noise shaping, which refers to a technique used where the quantization noise is pushed out of the frequency band. this is done by using a feedback loop that integrates the error between the input and output.

* How it works:-

- 1- the input analog signal is sampled at a much higher rate than the Nyquist rate, this is done to increase the resolution of the ADC and to reduce the quantization error.
 - 2- the oversampled signal is passed through a delta-sigma modulator. the modulator is a feedback loop that consists of a digital to analog converter (DAC) and a comparator. the DAC generates a reference voltage that is compared to the oversampled signal by the comparator. the output of the comparator is a 1-bit digital signal that has a high frequency pulse density effectively quantizes the signal with a high resolution.
 - the output of the delta sigma modulator is then passed through a digital filter that performs noise shaping. the digital filter removes the high frequency noise from the oversampled signal and shifts it to the higher frequency, where it can be easily filtered out. the output of the digital filter is a high-resolution digital signal that accurately represents the input analog signal.
 - 3- finally, the digital signal is processed by a microcontroller or a digital signal processor (DSP) to extract the desired information from the signal
- * Delta-sigma ADCs are widely used in Medical, audio applications where high-resolution and high-accuracy ADCs are required.

2) Flash ADCs (parallel ADCs) :-

→ is the fastest type of ADC and use large numbers of comparators.
 An N -bit flash ADC consists of (2^N) resistor and $(2^N - 1)$ comparators.



$$V_1 = \frac{R}{R+7R} * V_{ref} = \frac{1}{8} V_{ref}$$

$$V_2 = \frac{R+R}{8R} * V_{ref} = \frac{2}{8} V_{ref}$$

$$V_3 = \frac{3}{8} V_{ref}$$

$$V_7 = \frac{7}{8} V_{ref}$$

$$\text{resolution} = 1LSB = \frac{V_{ref}}{8}$$

- each comparator compares the input voltage with the reference voltage,
- if the input voltage is greater than the reference voltage, then the output of that particular comparator will become high, and if that is not the case, then it will remain Low.
- based on the comparator outputs, this encoder gives the binary code

→ Priority encoder: it determines the input with the highest priority and generates an output code that represents that input.

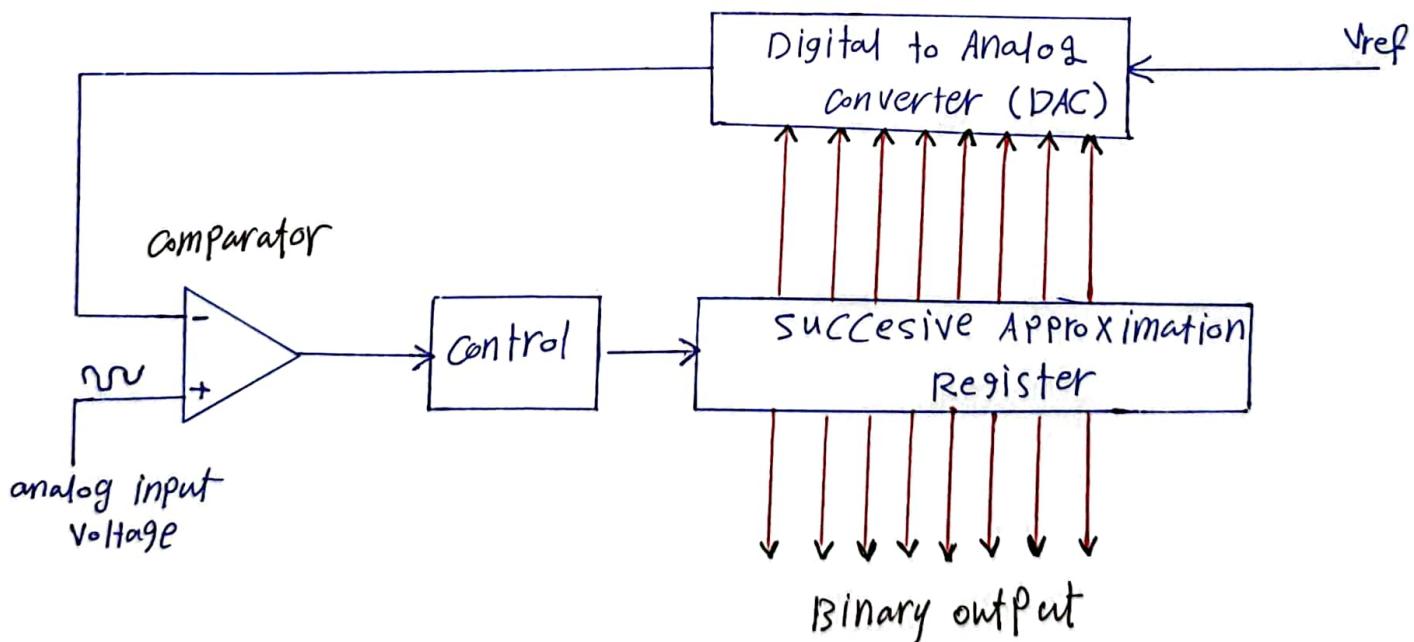
* Advantages :

- Fastest ADC
- suitable for large bandwidth applications
 - e.g. - satellite communication - Radar processing
 - oscilloscope
- catch every rapid change

* disadvantages:

- High power consumption
- limited resolution (8:12 bit)
- Large Die Area ($2^N - 1$ comparators) \Rightarrow High cost

3 - Successive Approximation ADC: (SAR)



- SAR → Successive Approximation Register
- middle-fast speed (< 5 MSPS) mega-samples-per-second
- middle resolution → 8:16 Bits
- this ADC applies a "binary search algorithm"
- the number of conversion steps is equal to the number of bits in the ADC

* How it works

- 1- the ADC samples the analog input signal and holds the value in a sample-and-hold circuit.
- 2- The ADC initializes a digital output value to zero and sets a most significant bit (MSB) to 1.
- 3- The ADC converts the current digital output value to an analog voltage and compares it with the input voltage. if the input voltage is greater than the digital output voltage, the corresponding bit is set to 1 otherwise, the bit is set to 0.
- 4- The ADC repeats step 3 for the next bit, continue from the (MSB) to the least significant bit (LSB)
- 5- after all bits have been determined, the ADC outputs the

remember → $D_{out} = \frac{V_{in}}{\text{step size}} = \frac{V_{in}}{(V_{ref}/2^{n-1})}$ *analog input voltage*

digital data output ↪ $\therefore V_{in} = D_{out} * \text{step size}$

ex: step size = 10 mV
input voltage = 1V

resolution → 8-bit SAR ADC

1- → MSB = 1

binary decimal	1000 0000 128
-------------------	------------------

→ $V_{in} = D_{out} * \text{step size} = 128 * 10 \text{ mV} = 1.28 \text{ mV}$
 $\therefore (1.28 > V_{in}) \rightarrow (1.28 > 1 \text{ V}) \rightarrow \text{clear bit}$

0100 0000 64

→ $V_{in} = 64 * 10 \text{ mV} = 640 \text{ mV} \rightarrow 640 \text{ mV} < 1 \text{ V}$
 $\therefore \text{bit 6 is kept since it smaller than } 1 \text{ V input}$

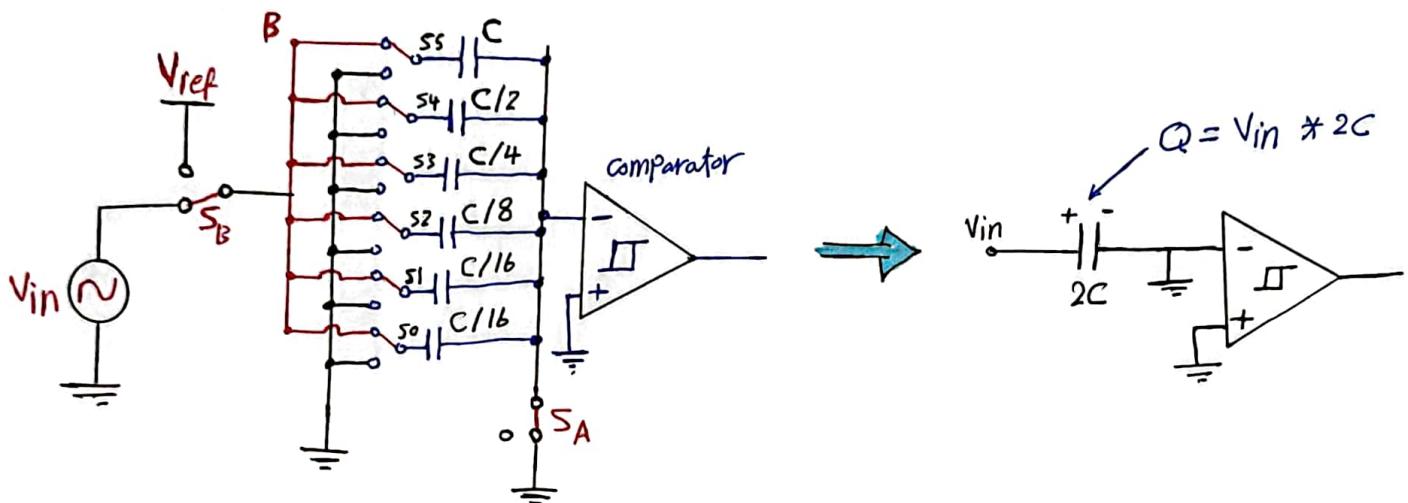
0110 0000 96

→ $V_{in} = 960 \text{ mV} < 1 \text{ V} \rightarrow \text{bit 5 is kept}$
 ... and so on

→ if the value of $V_{in} = 1 \text{ V}$ we let the bit = 1
 → the final result is 01100100 → 100 → 100 * 10 mV = 1V

- * Basic schematic of SAR switched-capacitor ADC ($\frac{5}{2}$ -bit ADC)
 - the capacitor values are binary weighted from C , $C/2 \dots C/2^{n-1}$ ($C \gg C/2 \dots C/16$), and the last two capacitors have the same value of $C/2^{n-1}$.
 - the conversion process is performed in three steps:-
 - 1 - sample mode
 - 2 - Hold mode
 - 3 - redistribution mode (the actual conversion)

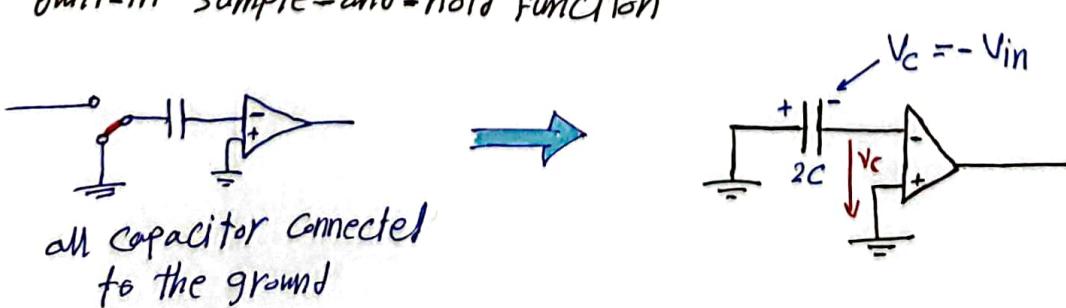
1 - Sample mode:-



- the bus switch S_B is switched to the input voltage (V_{in}), and S_A is closed to ground.
- the capacitor switches ($S_5 : S_0$) are turned to the common bus B and all capacitors are charged to V_{in}
- the total charge of $Q_{in} = V_{in} * 2C$ is stored on the capacitors.

2 - Hold mode:-

- Switch S_A is open while the switches ($S_5 : S_0$) are connected to ground, that way making the voltage at the inverting comparator input $V_C = -V_{in}$. This means that the switched-capacitor circuit already has a built-in sample-and-hold function



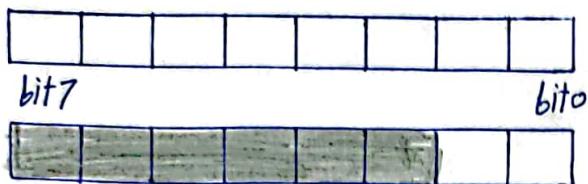
3- Redistribution mode :-



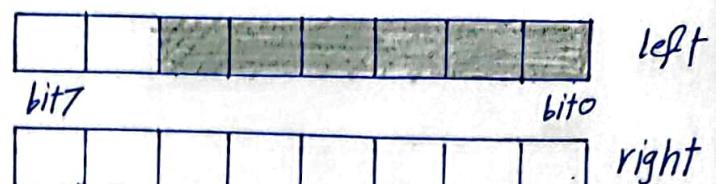
- the first conversion step, connects C (the largest capacitor) using switch S_5 to the reference Voltage (V_{ref}), which corresponds to the full-scale range (FSR) of the ADC, which refers to the maximum voltage range it can measure.
- Capacitor [C] forms a 1:1 capacitance divider with the remaining capacitors connected to ground. the comparator input Voltage becomes $V_C = (-V_{in}) + \frac{V_{ref}}{2}$. This Voltage (V_C) is compared with the potential Voltage of (0V) at the non-inverting input. therefore
 - if $\rightarrow |V_{in}| > \frac{V_{ref}}{2} \rightarrow V_C < 0 \rightarrow$ the comparator output \rightarrow high (1)
 - if $\rightarrow |V_{in}| < \frac{V_{ref}}{2} \rightarrow V_C > 0 \rightarrow$ the comparator output \rightarrow low (0)
- if V_0 (comparator output) \rightarrow High , switch S_5 remains connected to V_{ref} for the rest of conversion
- if $V_0 \rightarrow$ Low, S_5 is switched back to ground.
- This process is repeated for switches (S_4 to S_1), except (S_0) remains connected to ground for the entire conversion time.
- At each step, the charge on the capacitor is redistributed based on the comparator output and the switch connections. This determines the value of the corresponding bit.
- After all bits have been determined, the converted digital value represents the analog input voltage (V_{in})
- this method does not require a DAC and uses only capacitors and comparators to perform the analog to digital conversion.

* ADC result formatting :- (10-bit)

→ the 10-bit A/D conversion can be supplied in two formatting:-
 - Left justified - right justified



ADC high register



ADC low register

* Calculate the ADC result in decimal format.

→ it looks like a normal conversion from binary format to decimal format:-
 ex:

the result = 11 1111 1111 → right justified

$$\text{ADC low} = 1111 1111 = 255$$

$$\text{ADC high} = 11 \rightarrow = 3 \rightarrow 3 \ll 8 = 3 * 2^8 = 768$$

$$\text{ADC result} = \text{low} + \text{high} = 255 + 768 = 1023$$

So:

$$\text{the ADC Value} = \text{ADC_low} + (\text{ADC_High} * 255) \quad \overbrace{2^8}$$

LM35 Temperature sensor interfacing

→ LM35 is a temperature sensor that can measure temperature in the range of -55°C to 150°C

→ it is a 3-terminal device that provides an analog voltage proportional to the temperature

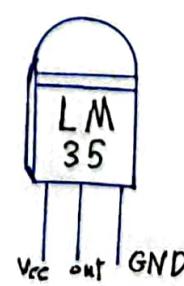
→ the higher the temperature, the higher is the output voltage.

→ the output analog voltage can be converted to digital form using ADC so that a microcontroller can process it.

→ V_{cc}: supply voltage (4V : 30V)

→ The sensitivity of LM35 is 10mV / degree celsius. As temperature increases, output voltage also increases.

$$\text{e.g. } \rightarrow 250 \text{ mV} \rightarrow \frac{250 \text{ mV}}{10 \text{ mV}} = 25^{\circ}\text{C}$$



→ assuming that the $V_{cc} = 5V$ and we have a 10-bit A/D system
 so::

$$\text{step-size} = \frac{V_{ref}}{2^{10}} = \frac{5}{1024} = 4.88 \text{ mV}$$

$$\text{analog-Voltage} = \text{step-size} * \text{ADC-Result}$$

ex :

ADC-Res Voltage

0

0 V

1023

5 V

256

X

$$\rightarrow X = 256 * \frac{5}{1023} = 1.251$$

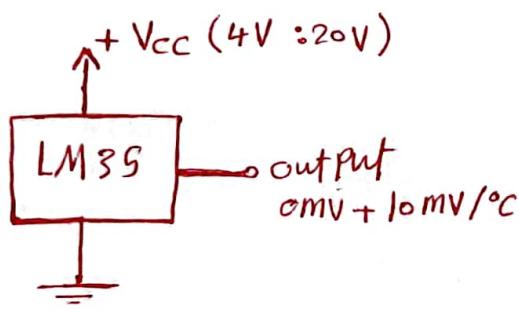
[X] → the equivalent voltage to the ADC-result

* the output voltage range of LM35

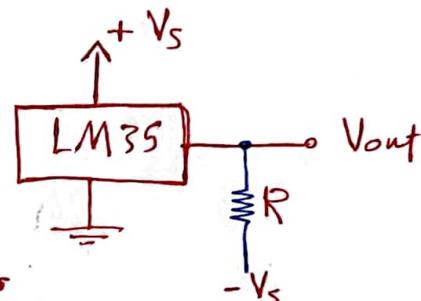
→ we have 2 basic connections for LM35

→ basic temperature measure setup

→ full range temperature measure setup



$$R = -\frac{V_s}{50mA}$$



$$\begin{aligned} V_{out} &= 1500 \text{ mV} \rightarrow 150^\circ \text{C} \\ &= 250 \text{ mV} \rightarrow 25^\circ \text{C} \\ &= -950 \text{ mV} \rightarrow -55^\circ \text{C} \end{aligned}$$

$$V_{out} = \text{step-size} * \text{ADC-result}$$

$$\text{temperature} = V_{out} / 10mV$$



* How to improve the resolution of ADC ?

- 1- oversampling : take multiple samples and average them together. this can effectively increase the resolution of the ADC by the number of samples taken. for example, if an ADC has a resolution of 10 bits and you oversample by a factor of 10, the effective resolution of the ADC will be 100 bits
- 2- Averaging : take multiple measurements of the same signal and average them together.
 - this can effectively increase the resolution of the ADC, but it is not as effective as oversampling
 - for example, if an ADC has a resolution of 10 bits and you average 10 measurements together, the effective resolution of the ADC will be 10.2 bits
- 3- Decrease step size : reduce the change in analog voltage that corresponds to a single bit change , effectively increasing the resolution because smaller changes in analog voltage can be detected.
- 4- Digital Post-processing : using DSP techniques which include interpolation , decimation, reconstruction
 - interpolation : adding new samples between the existing samples
 - decimation : removing samples from the signal
 - reconstruction : reconstructing the signal from the samples.
- * interpolation and decimation can increase the resolution of an ADC by a factor equal to the number of new samples added or removed.
- * reconstruction : can increase the resolution by the factor ~~is~~ equal to the number of samples used to reconstruct the signal.
- oversampling and averaging can increase the resolution of an ADC
- digital-post-processing can further enhance the ADC's performance.
- oversampling is the most effective way to improve resolution, but it can also be expensive .
- Averaging is a less expensive option, but not as effective as oversampling
- the best method depends on the application and available resources .

Steps for using the ADC to perform an A/D conversion :-

1- Configure port :-

- disable pin output driver
- Configure pin as analog (disable the input buffer)

2- Configure the ADC module :-

- select ADC conversion clock
- Configure voltage reference
- select ADC input channel
- select result format
- select Acquisition delay
- turn on ADC module

3- Configure ADC interrupt (optional)

- clear ADC interrupt flag
- enable ADC interrupt
- enable peripheral interrupt
- enable global interrupt

4- Wait the required acquisition time

5- Start the Conversion

6- Wait for ADC conversion to complete by one of the following :

- Polling
- waiting for the ADC interrupt

7- Read ADC result

8- Clear the ADC interrupt flag (required if interrupt is enabled)

