ANALOG – TO – DIGITAL CONVERTER

An **analog-to-digital converter** (**ADC) converts an analog signal into a digital code.**

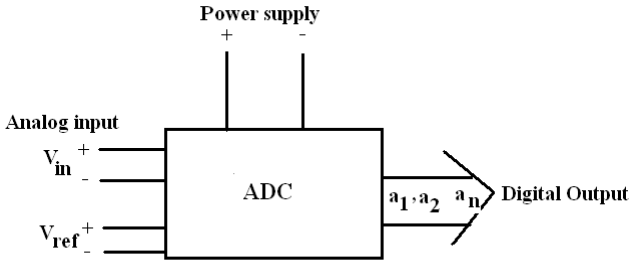


Fig. Block Diagram of ADC

The input/output relationship is given by the equation

$$\frac{V_{in}}{V_{ref}} = a_1 2^{-1} + a_2 2^{-2} + \ldots\ldots\ldots + a_j 2^{-j} + \ldots\ldots\ldots upto \ \infty$$

Practical ADC can have only a finite number of bits (say *n*)

$$\frac{V_{in}}{V_{ref}} = a_1 2^{-1} + a_2 2^{-2} + \ldots\ldots\ldots + a_j 2^{-n}$$

$$\sum_{i=1}^{n} a_i 2^{-i} = \frac{V_{in}}{V_{ref}}$$

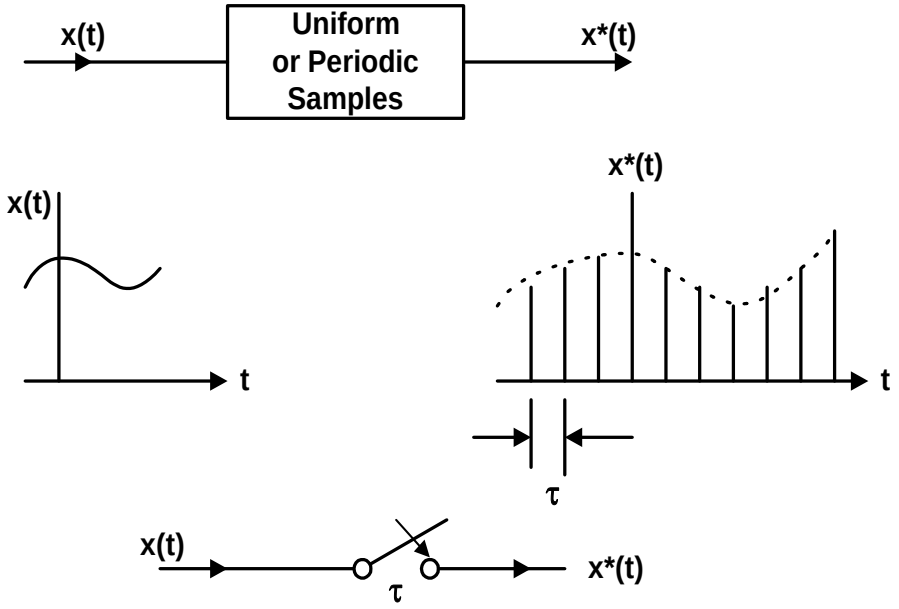
Where V_{in}= Unknown voltage,

V_{ref} = reference voltage = V_{fs} *(in most cases)*

The overall process of converting an analog signal to a digital form involves:

1. **SAMPLING**
2. **HOLDING**
3. **QUANTIZING**
4. **ENCODING**

SAMPLING



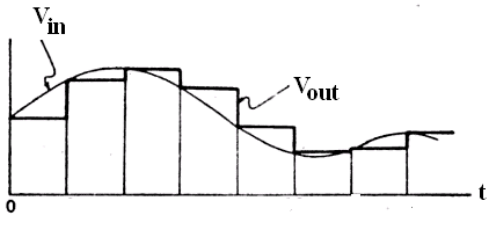
Ideal switch (closes periodically at intervals of τ, remains closed for a moment and opens immediately).

τ is the sampling period

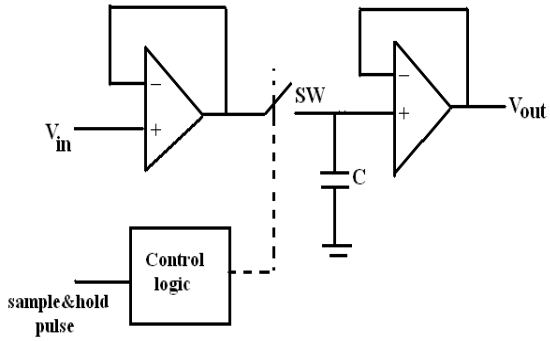
$f_s = \frac{1}{\tau}$ is the sampling frequency.

HOLDING

"Holding" operation means storing or preserving the selected sample value until its conversion to digital form is over.



"Sampling" & "Holding" are performed by Sample & Hold Circuit (as shown below)



Quantizing

Practical ADC consists of finite number of bits.

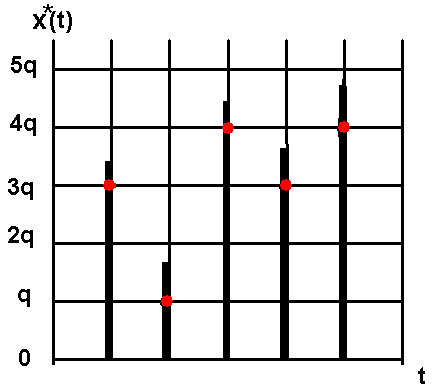
So, digital word can assume only certain number of discrete values. Hence, *quantization* in the amplitude of the samples takes place.

In quantization, the full scale range of the ADC is partitioned into 2^n segments (called a *quantum* or *quantization step* '*q*') each of width, $q = \frac{V_{FS}}{2^n}$.

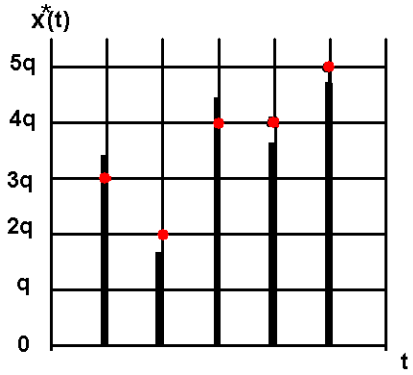
Any analog voltage $E < V_{FS}$ is replaced by a discrete voltage which is an integral multiple of the quantization step [i.e. 0, q, 2q, 3q, ..., $(2^n - 1)q$].

This may be realized by *truncation* or *rounding off*.

However, almost all commercially available ADCs are *rounding- off* type.



Truncation type



Rounding off type

Encoding

Expressing digital words by suitable binary code.

Basic A/D converters

- Counter-ramp Converter
- Successive Approximation Converter
- Dual Slope Converter

COUNTER-RAMP CONVERTER

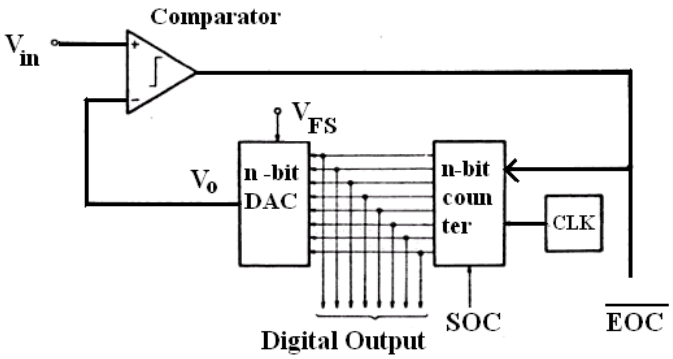
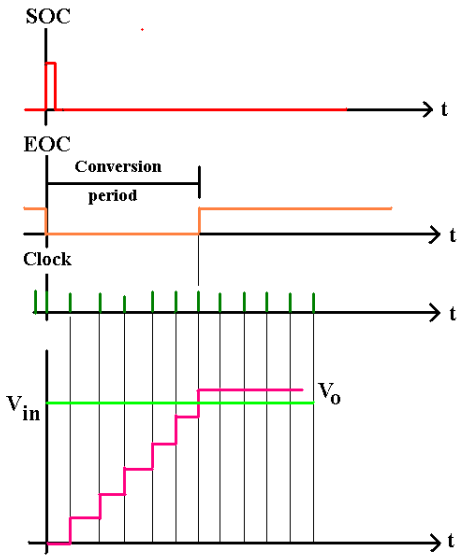


Fig. Counter Type ADC

- At the beginning of the conversion, the counter is set to zero.
- SOC goes high, counter starts counting.
- Allow the converter to increment until the output of the DAC exceeds the analogue input when the output of the comparator changes and makes EOC high.



Disadvantages

- **Variable conversion time.**

SUCCESSIVE APPROXIMATION CONVERTER

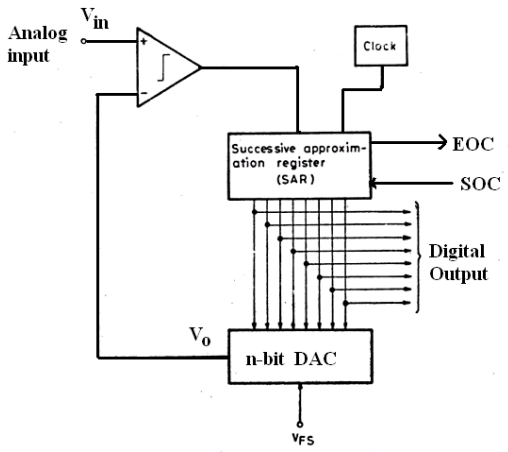
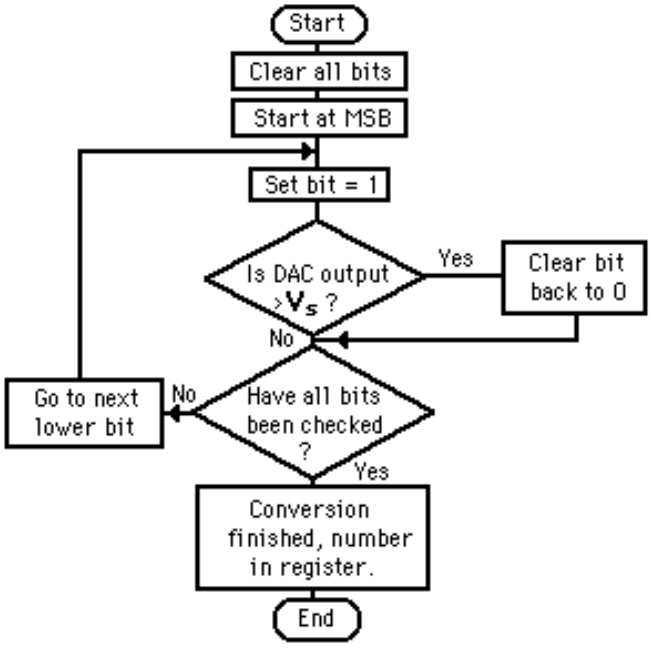


FIG. SAR Type ADC

- Successive approximation works by constantly comparing the input voltage to the output of an internal digital to analogue converter (DAC) until the best approximation is achieved.
- At start, the successive approximation register is initialized so that the most significant bit (MSB) is set to a logic 1. This code is fed into the DAC which then supplies the analog equivalent of this digital code ($V_{ref}/2$) into the comparator circuit for comparison with the input voltage.
- If this analogue voltage exceeds V_{in} the comparator causes the SAR to reset that bit and If it is lower, then the bit is left a 1. Then sets the next bit to logic 1. This binary search continues until every bit in the SAR has been tested.
- The resulting code is the digital approximation of the sampled input voltage and is finally output by the ADC at the end of the conversion (EOC).



Advantages

- **Faster in operation**
- **Constant conversion time (= n / f_c) and only dependent on clock.**

DUAL SLOPE CONVERTER

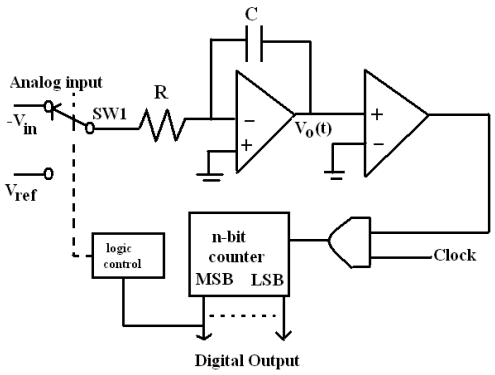


Fig. Dual Slope ADC

- In operation the capacitor of the integrator is discharged and the counter is set to zero count.
- The input to the integrator then attached to the analog input (SW1 up).The output of the integrator will be linearly rising ramp.
  Thus after conversion process starts the comparator output switches to HIGH state, which starts counter.
- When MSB of the counter becomes logical 1(at time $t=t_a$), the output voltage will be

$$V_0(t_a) = -\frac{1}{RC}\int_0^{t_a}(-V_{in})dt = \frac{V_{in}}{RC}t_a$$

Then input of the integrator switches to V_{ref} (SW1 down). The output of the converter starts decreasing linearly.
At time $t=t_b$, the output of the integrator becomes zero which changes the output of the comparator to a LOW state. This stops the counter.

- The output of after time $t=t_b$ is given as:

$$V_0(t_b) = \frac{V_{in}}{RC}t_a - \frac{1}{RC}\int_{t_a}^{t_b}(-V_{ref})dt = 0$$

$$0 = \frac{V_{in}}{RC}t_a - \frac{V_{ref}}{RC}(t_b - t_a)$$

From above

$$V_{in} = \frac{(t_b - t_a)}{t_a} V_{ref}$$

If f_c be the frequency of the clock pulses, the time required for counter to set the MSB logical 1 is given as

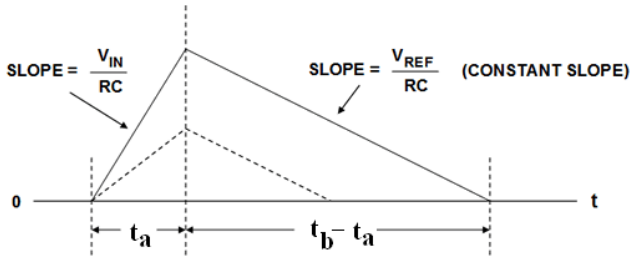
$$t_a = \frac{2^{n-1}}{f_c}$$

If the number of counts *M* which accumulate in counter before the integrator reaches zero volts output and the comparator output changes, then

$$t_b - t_a = \frac{M}{f_c}$$

$$V_{in} = \frac{M}{2^{n-1}} V_{ref}$$

$$\propto M \; as \; \frac{V_{ref}}{2^{n-1}} = constant$$



Advantages

- Conversion accuracy is independent of RC value, provided RC remain constant during conversion period.
- Conversion accuracy is independent of clock frequeny fc as long as it remain constant during conversion period.

Disadvantages

- Variable conversion period

Quantization Error

In natural binary code

$$N = b_1 2^{-1} + b_2 2^{-2} + \cdots \ldots b_i 2^{-i} + \cdots . upto \propto \ldots..(1)$$

Since the ADC can have only a finite number of bits (say B)

$$N = b_1 2^{-1} + b_2 2^{-2} + \cdots \ldots \ldots b_B 2^{-B}$$

Therefore

$$V_{in} = V_{ref}(b_1 2^{-1} + b_2 2^{-2} + \cdots \ldots \ldots b_B 2^{-B})$$

The truncation of the number of terms on the R.H.S. of expression(1) represents quantization operation.

Comment on full scale:

The maximum possible quantized value of the input voltage is $\frac{7}{8} V_{ref}$ **for a 3 bit ADC. Hence for a B- bit ADC, it is,**

$\frac{(2^B - 1)}{2^B} V_{ref}$**. Hence the maximum possible quantized value is less than the full scale range by** $V_{ref}/2^B$ **i.e. one quantization step, i.e. 1 LSB.**

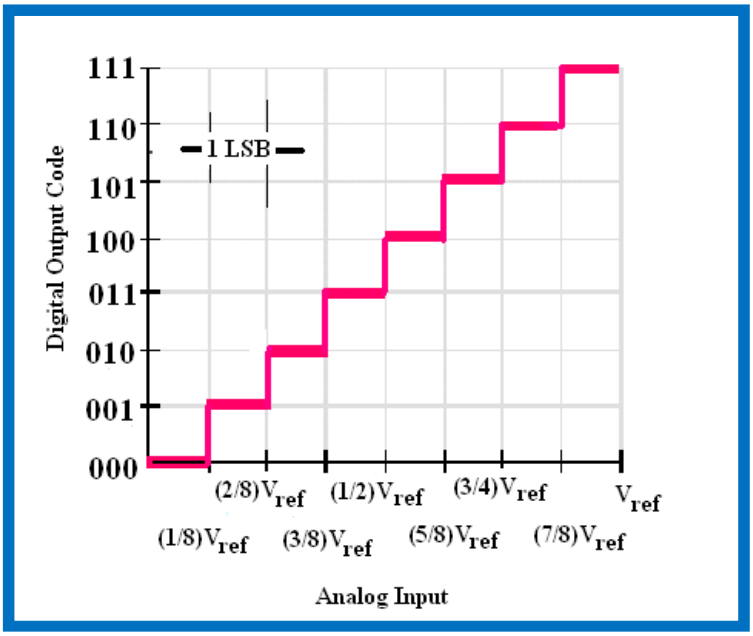
The maximum error we have here is 1 LSB. This 0 to 1 LSB range is known as the "quantization uncertainty or ***"quantization error"*****.**

This error results from the finite resolution of the ADC. i.e., the ADC can only ***resolve*** **the input into** 2^B **discrete values.**

Each output code represents a range of input values. This range of values is a ***quanta,*** **to which we assign the symbol** ***q*****.**

The converter resolution, then, is 2^B.

So, for an 8 Volt reference (with a unity gain factor), a 3-bit converter *resolves* the input into $V_{REF}/8 = 8/8 = 1$ Volt steps.



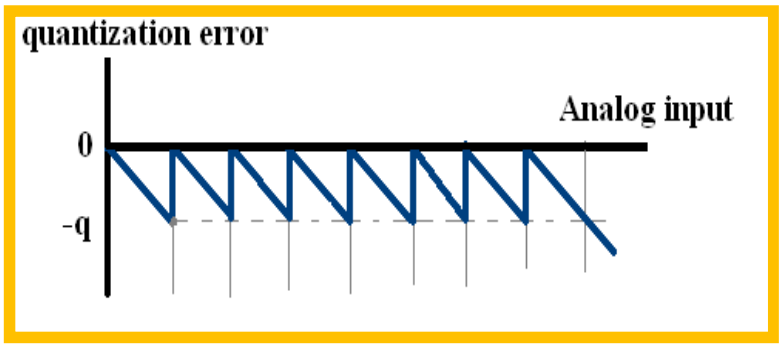


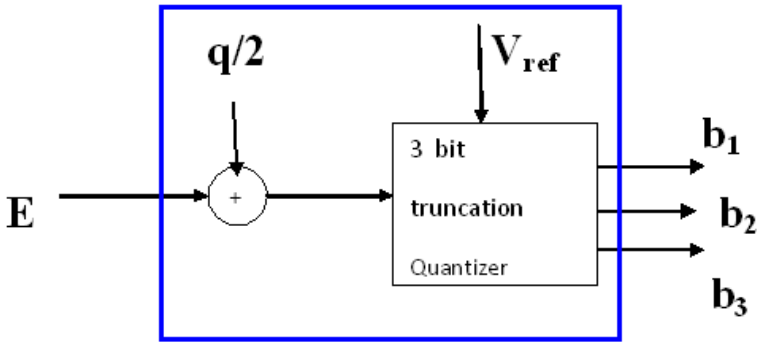
Fig. Input/Output relatonship of 3-bit Truncation type ADC

The quantization error (or uncetainty), *e* is given by

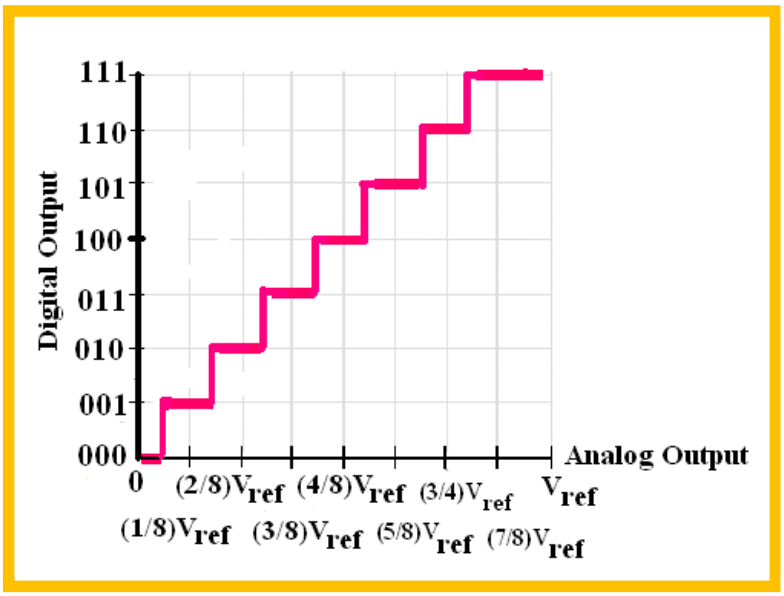
***e* =quantization error = Quantized value – Actual value**

From the curve, $|e_{max}| = q$, the quantization step, the value corresponding to 1LSB.

$|e_{max}|$ can be reduced to q/2 if an offset voltage of q/2 is added prior to quantization using truncation. This is equivalent to rounding off type quantization.



Here ouput changes state at the midpoint of each quantization step(except last step).



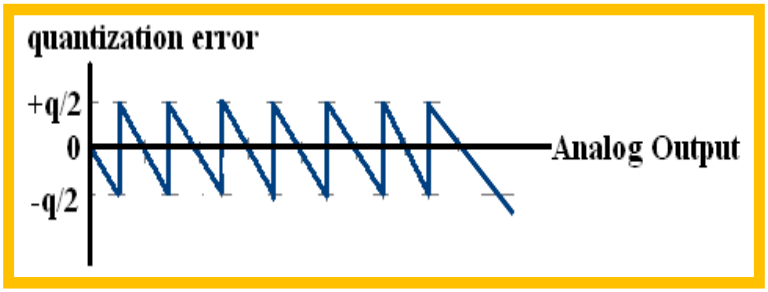


Fig. Input/Output relationship 3-bit rounding off type ADC

ADC ERRORS

Offset Error

The offset error is defined as a common deviation of the transition voltages from their ideal values.

- ✓ **Offset error results in a horizontal transition of the graph for input/output relation.**
- ✓ **Offset error arise in ADC because of amplifier and comparator input offset voltages.**

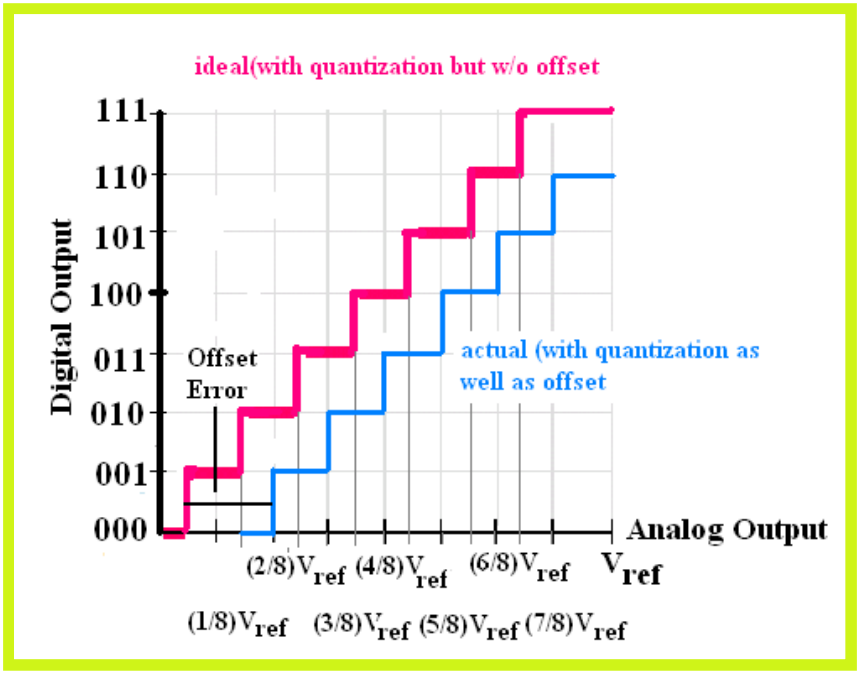


Fig. ADC with offset Error

- ❖ **There is provision for offset compensation by the user at a particular ambient temperature. However due to the temperature dependence of the offset, the compensation is not valid at other temperatures.**

Scale factor error or Gain Error

The gain error of an ADC is defined as the change in slope of the ideal transition curve.

For ideal input/output relation

$$N = \frac{V_{in}}{V_{ref}}$$

- ✓ **The gain $\frac{1}{V_{ref}}$ *or* $\frac{1}{V_{fs}}$ can change due to change in V_{ref} or due to change in resistor values and amplifier gain in the internal circuitry of ADC. i.e., N= $\frac{V_{in}}{\alpha V_{\mathrm{Re}\, f}}$ where α is a +ve coeff. which is 1 in absence of gain error & either < or > 1 in presence of gain error.**
- ✓ **With increase in slope, the quantization step becomes smaller and final output code is reached before the ideal transition voltage is reached.**
- ✓ **With decrease in slope the quantisation step becomes larger**

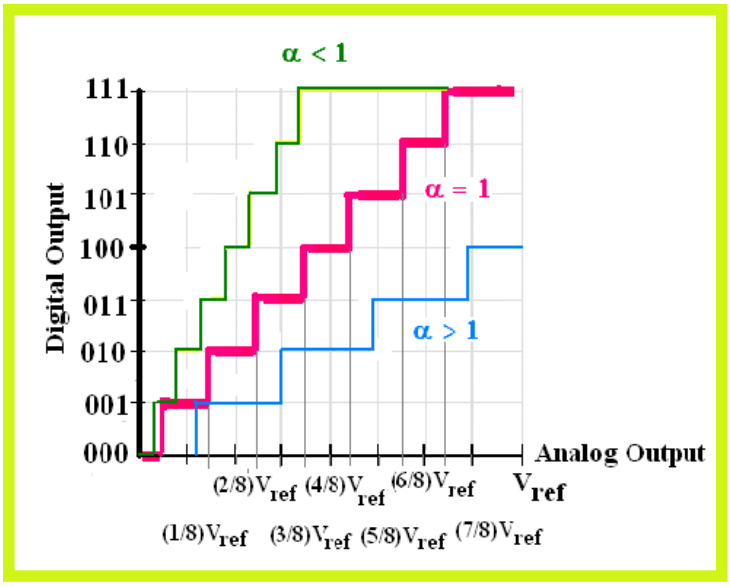


Fig. ADC with Gain Error

How to measure Gain error?

- **In a rounding off type ADC where only quantization error is present, 1st transition voltage is ½ *LSB* & last transition voltage is V_{FS} – *1½ LSB.***
- **Hence difference betn last & 1st transition voltages is V_{FS} – *2 LSB***
- **Then as a convention, gain error is obtd as ,**

G.E. = (Ideal difference between last & 1st transition voltages)

- (Measured difference between last & first transition voltages)

Gain error can be expressed in terms of LSB or as % of V_{FS}.

Linearity Error

The linearity error in ADC is defined as the deviation of code widths from their nominal values and that too by unequal amounts for different codes.

Linearity error can be expressed in two ways :

- **Differential Non-linearity (DNL) : It is defined as the difference between the actual code width and the ideal code width.**

  DNL = Actual code width – ideal code width

  In wrost case if DNL becomes large then certain codes may vanish i.e. Missing code.

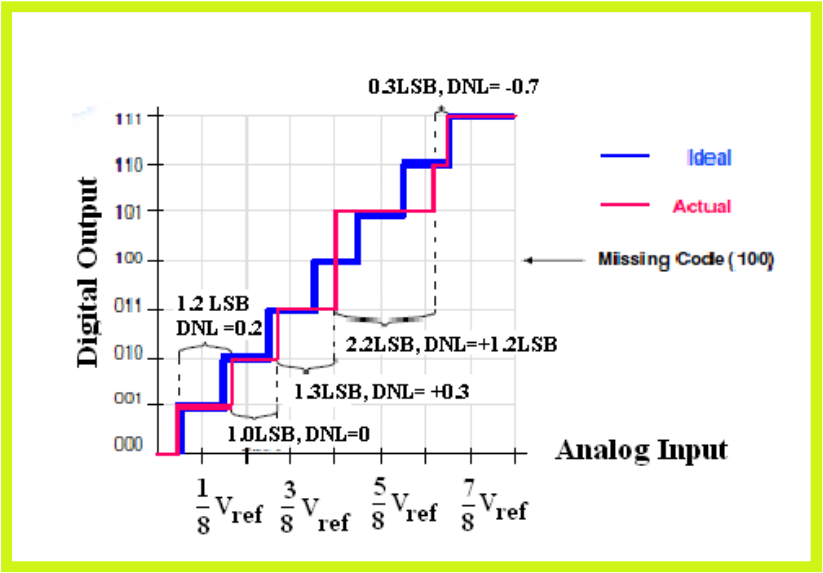


Fig. ADC with Linearity Error

- **Integral Non-Linearity(INL): The INL is defined as the deviation of the actual ADC transition from an ideal transition drawn through the first and last code transition**.

$$INL = \sum DNL$$

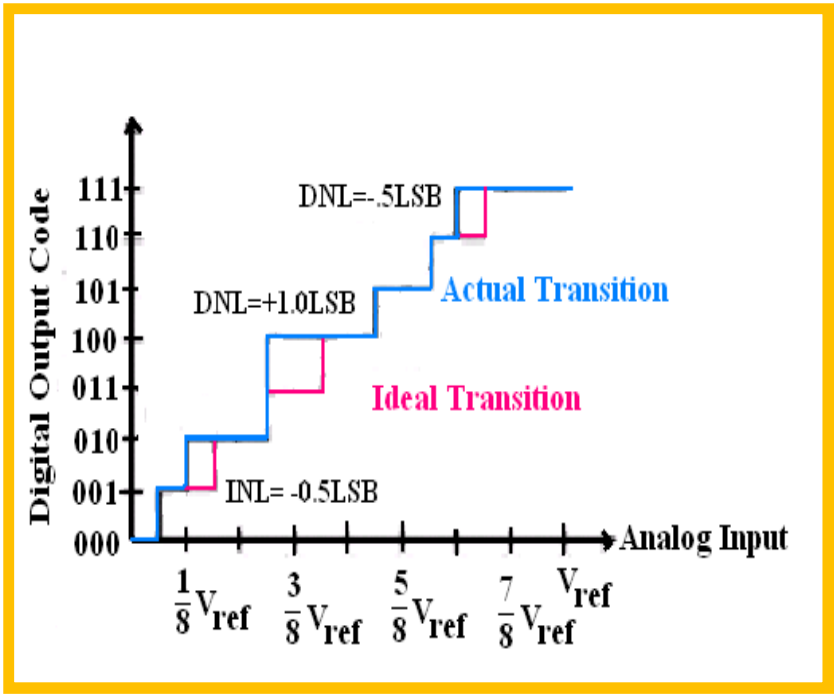


Fig. ADC with INL Error