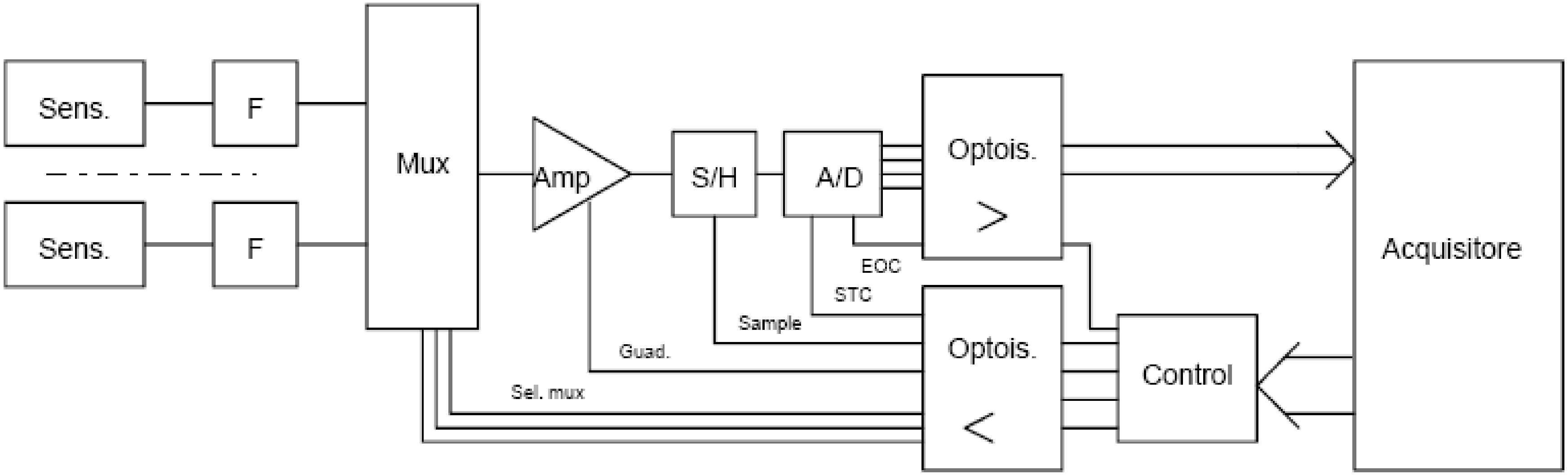
ANALOG INTERFACING

ANALOG INTERFACING

Analog signals = they provide the physical magnitude measurement through an electrical correlated one (usually V o I, but also impedance variation)



The acquisition unit actuates the following operations:

- **device activation times and synchronization**
- **Anti drift calibration**
- **Perturbing magnitude estimation (influence on the measurement)**
- **derived measurement calculation**
- **acceptable values/diagnostic**

ANALOG INTERFACING

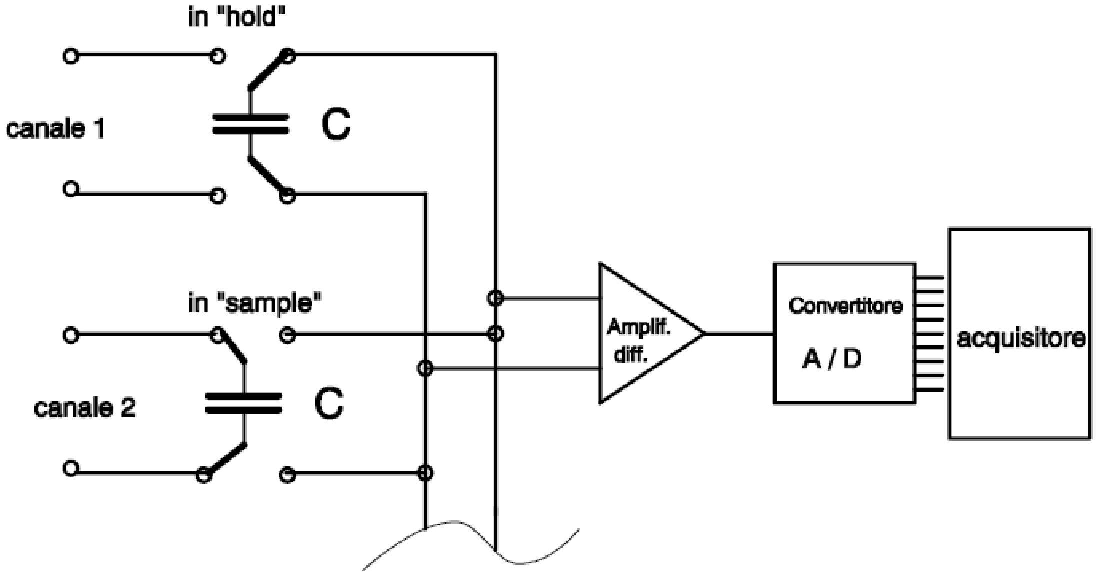
Filtering:

- low pass or band pass filter to eliminate the aliasing due to sampling

- zero mean components filtering is performed by the acquisition unit (μP)

- datasheet: Z_{in}, Z_{out}, cutoff frequency, frequency transfer function

Multiplexer:



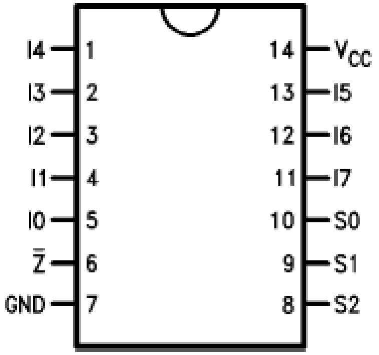
- ❑**1 ADC for N signals**
- ❑**< costs,**
- ❑**< f_c, no simultaneous samples**
- ❑**Mux "fly capacitor"**
  - **t_{comm} = msec**
  - **vita = 10^6 commutations**
  - **Galvanic insulation (S&H embedded)**
  - **high Z_{in} op. amp. to avoid capacitor discharge during conversion**

ANALOG INTERFACING

Semiconductor Multiplexer:

- t_{comm} = μsec
- life = 10^9 commutations
- No galvanic insulation
- Small size

Dual-In-Line Package

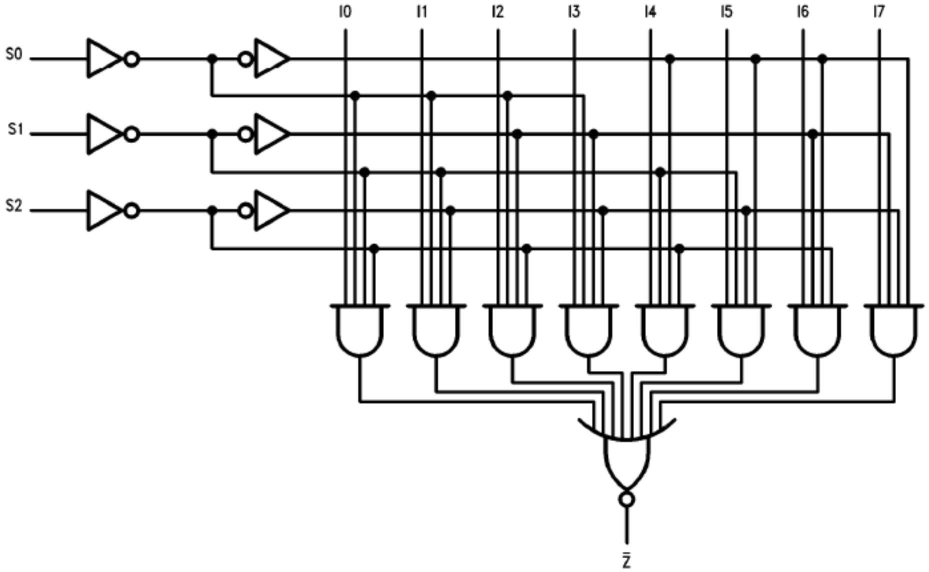


TL/F/10206–1

Order Number 54LS152FMQB
See NS Package Number W14B

Pin Names	Description
I0–I7 S0–S2 $\overline{Z}$	Data Inputs Select Inputs Inverted Data Outputs

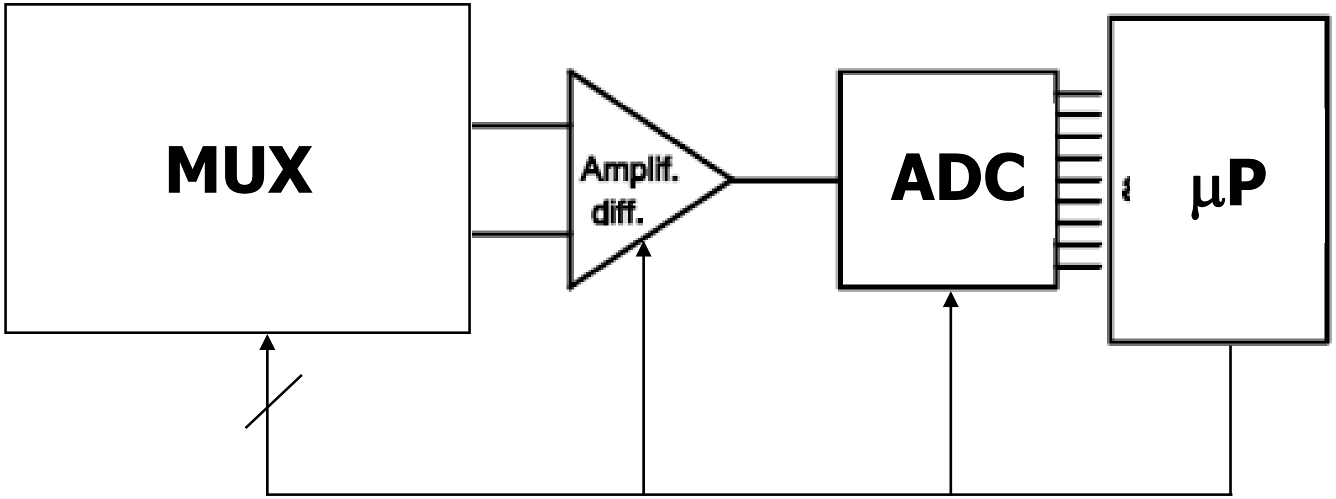
Logic Diagram



ANALOG INTERFACING

multiplexer characteristics:

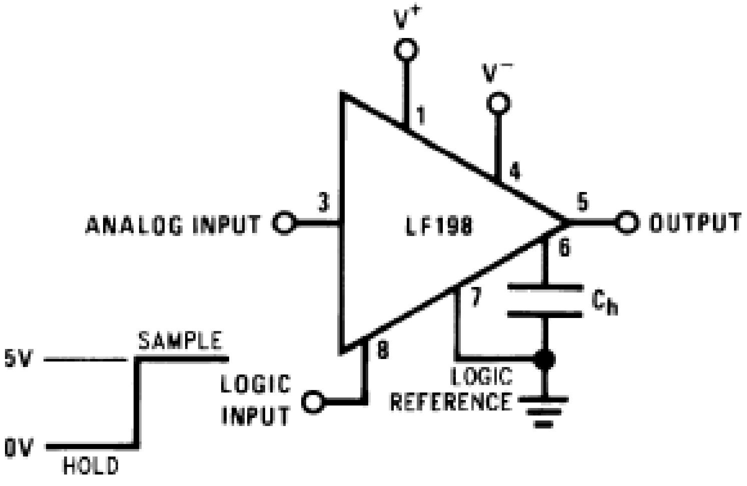
- **Number of channels**
- **Commutation speed (msec/μsec)**
- **R_{on} e R_{off} (open/close circuit resistances)**
- **S&H command from the μP must have enough duration to sample, consider also conversion time of the ADC and settling time of the operational amplifier**



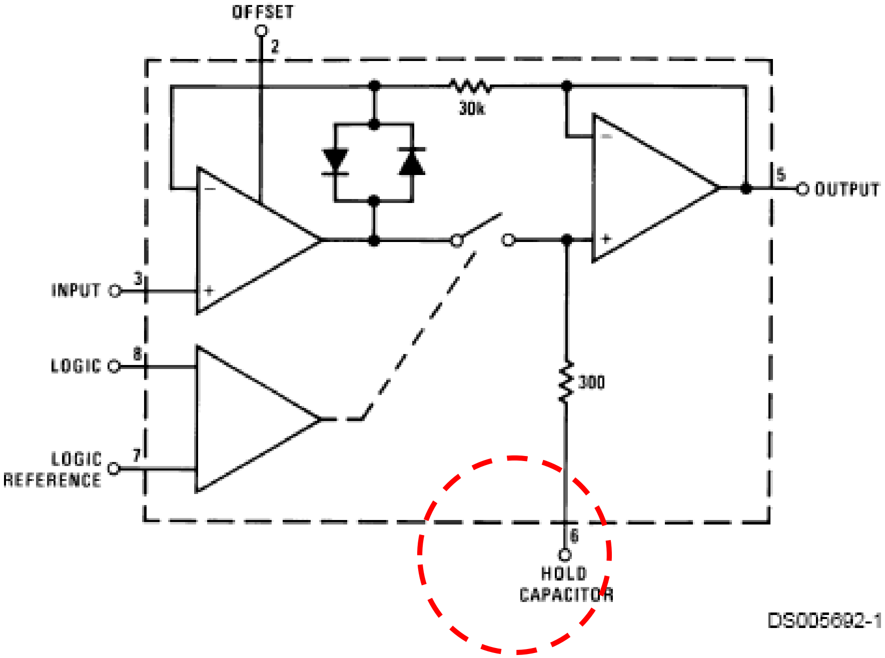
ANALOG INTERFACING

Sample and Hold:

- **Single integrated in a component triggered by a μP**
- **Input circuit to de-couple signal and command**
- **Output stage with operational amplifier to de-couple successive stages**
- **It makes the signal stable for the subsequent ADC conversion (not always necessary)**



DS005692-32

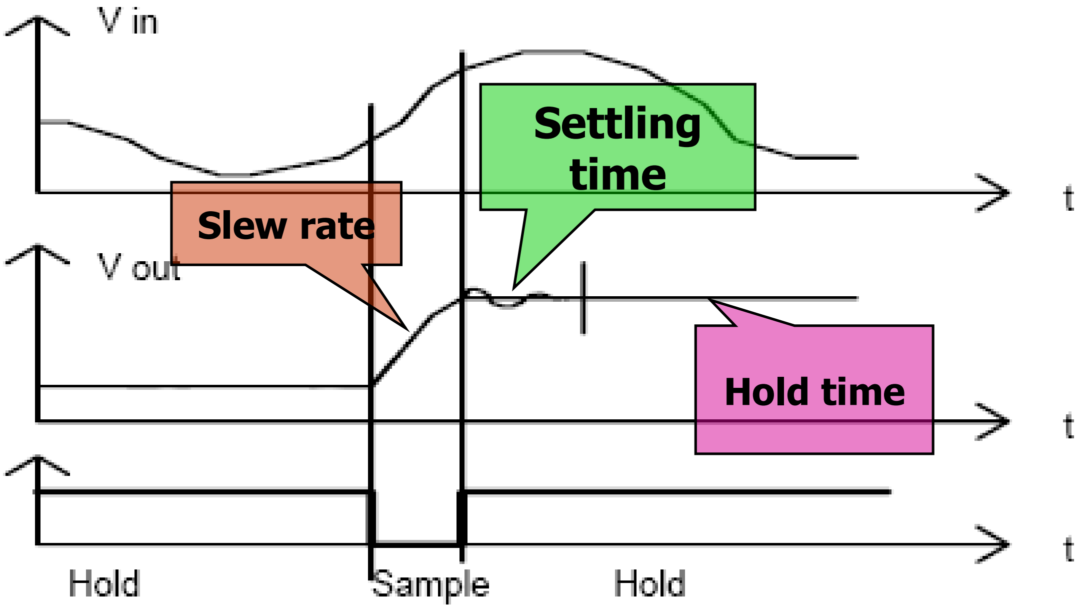


DS005692-1

ANALOG INTERFACING

Sample and Hold, characteristics:

- Z_{in} /Z_{out}
- Slew rate
- Settling time
- Hold time



Input Signal

Output signal

Command

Slew rate

Settling time

Hold time

ANALOG INTERFACING

Operational Amplifier:

- The signal from the transducer must be amplified to bring it in the range of the ADC
- De-coupling (Z_{in} e Z_{out})
- Bandwidth and slew rate
- The Op-Amp is not always necessary if present in other components of the acquisition chain
- Programmable (variable) gain by a μP
- Calibration for offset compensation
- Its settling time influences the synchronization of the μP commands

ANALOG INTERFACING

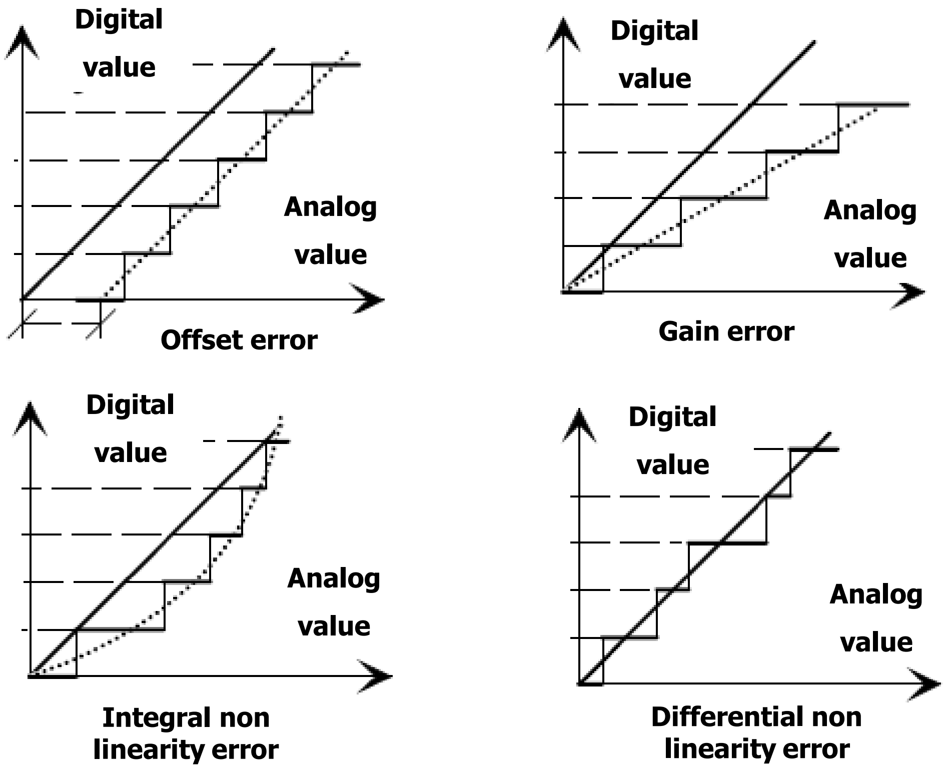
Analog to Digital Conversion (ADC):

The signal from the transducer can be read by the μP in a binary form

- **Input: range of V_{in} [0…5V], [-5V…5V], [0…10V], V_{rif}=10V**
- **Output: 8-10-12-14-16-24 bit values; fixed point with or without sign**
- **Characteristics: conversion approach, costs, velocity, accuracy**
- **Quantization error constant in absolute value ⇒ the role of the amplificier is important to exploit the input range of the ADC**
- **Conversion techniques based on:**
  - **Closed loop comparison: DAC + comparison**
  - **Time evaluation of the charge of a capacitor or of its charge/discharge frequency**
  - **Direct comparison through resistive networks voltage levels**
- **Calibration for offset compensation**
- **Influence on synchronization of the μP commands due to conversion time.**

ANALOG INTERFACING

Conversion errors:



Offset error

Gain error

Integral non linearity error

Differential non linearity error

Errors due to non ideal ADC

Offset error: the transfer function does not cross the zero point (axes origin)

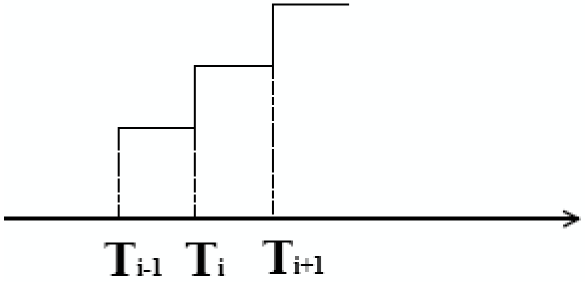
Gain error: the slope is ≠ from 45° (constant step amplitude but ≠ 1 LSB)

Integral Non Linearity Error: the “overall” curve is not linear

Differential Non Linearity Error: variable step amplitude and ≠ 1 LSB

ANALOG INTERFACING

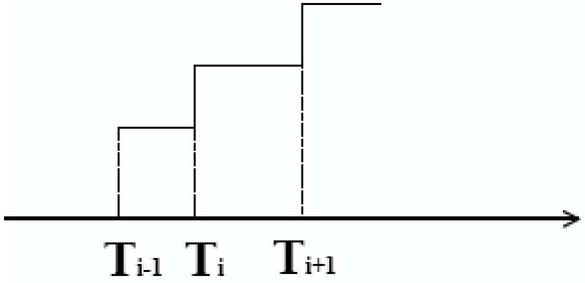
Differential non linearity error (DNL):



Ideal case

$T_{i+1} - T_i = \Delta =$ Nominal quantization step

$$\frac{T_{i+1} - T_i}{\Delta} = 1 \Longrightarrow \frac{T_{i+1} - T_i}{\Delta} - 1 = 0$$



Real case

$T_{i+1} - T_i \neq \Delta =$ Nominal quantization step

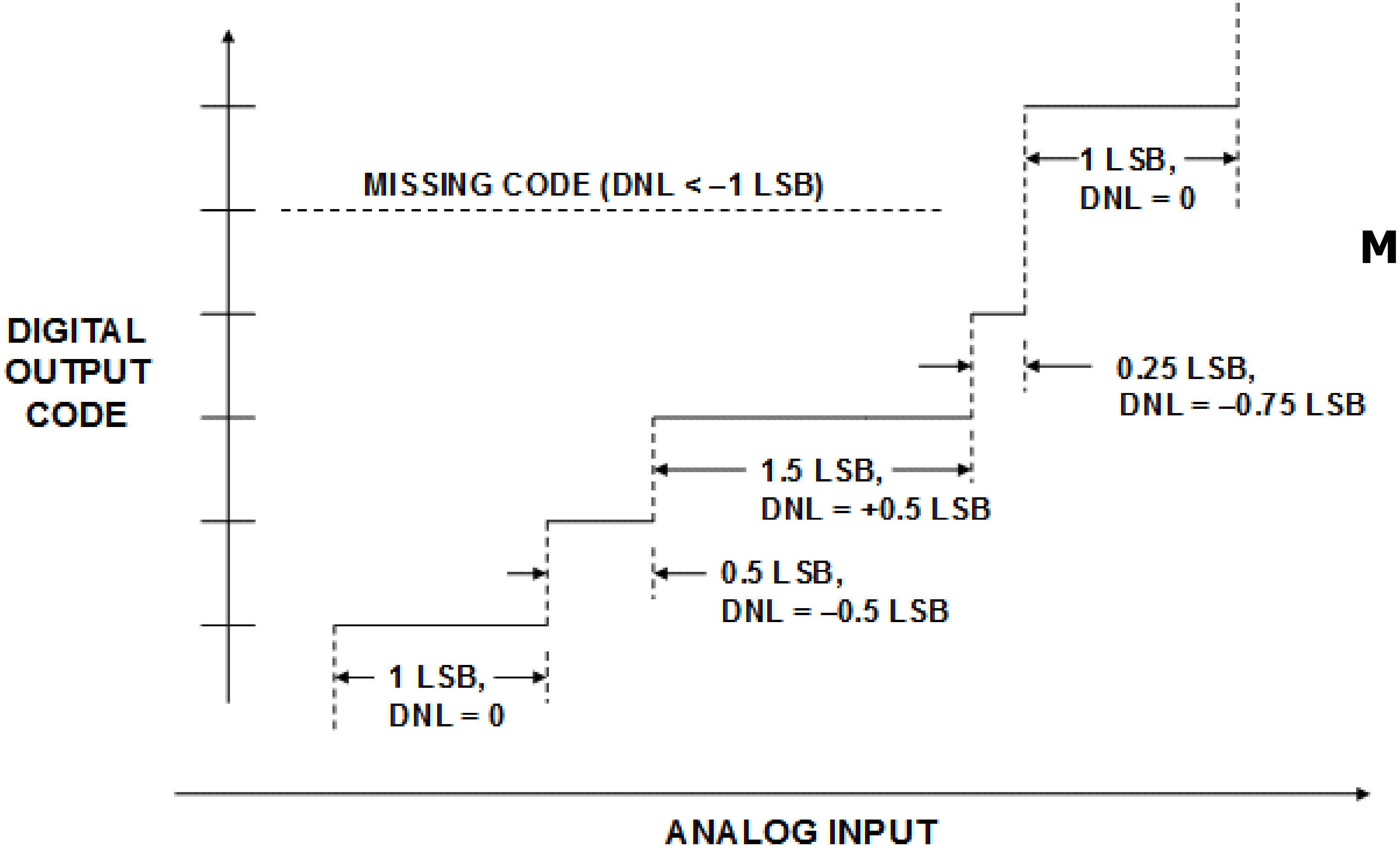
$$\frac{T_{i+1} - T_i}{\Delta} \neq 1 \Longrightarrow \frac{T_{i+1} - T_i}{\Delta} - 1 \neq 0$$

$$DNL_i = \frac{T_{i+1} - T_i}{\Delta} - 1$$

Differential non linearity referred to the ith step

ANALOG INTERFACING

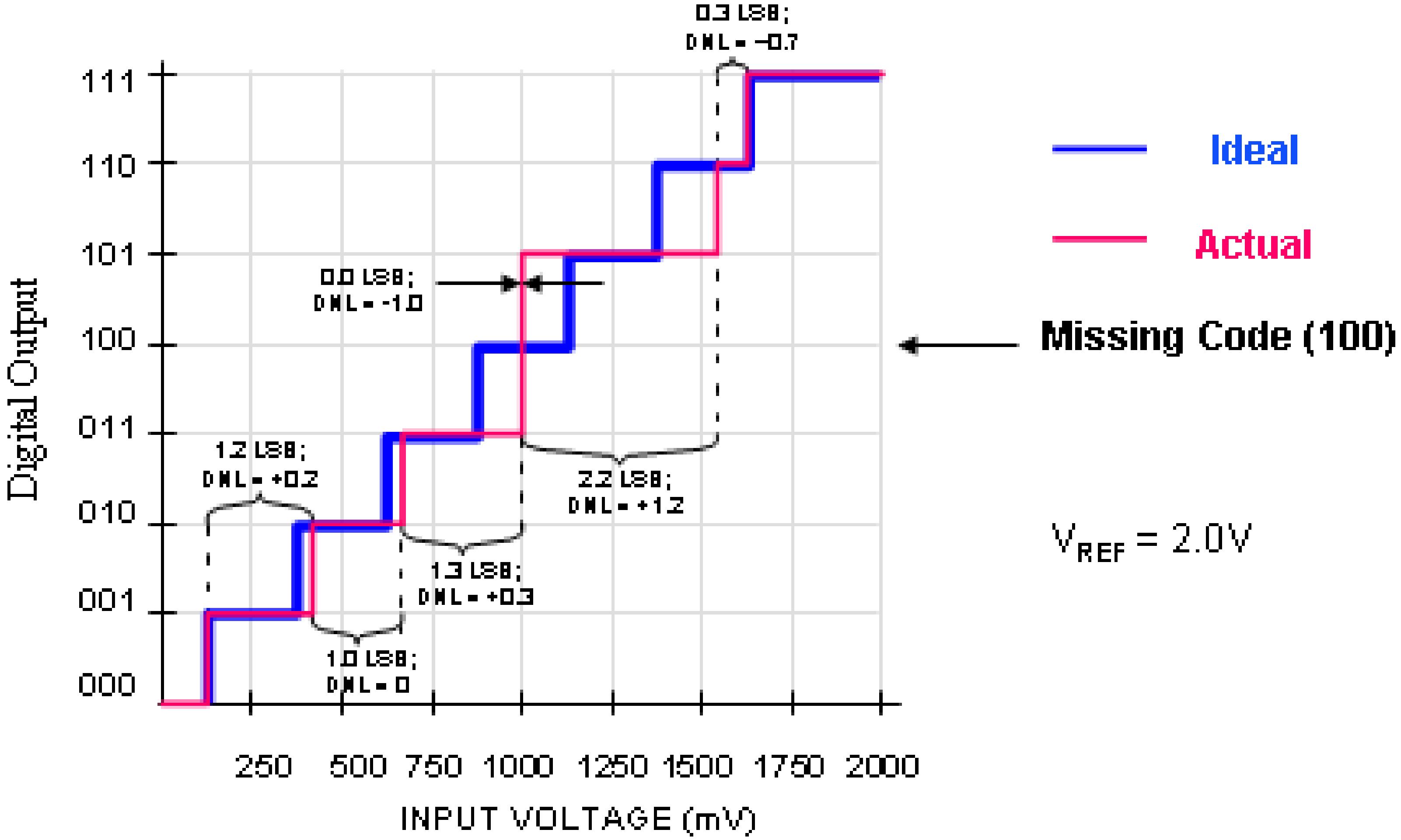
Differential non linearity error (DNL) for an ADC:



Missing code

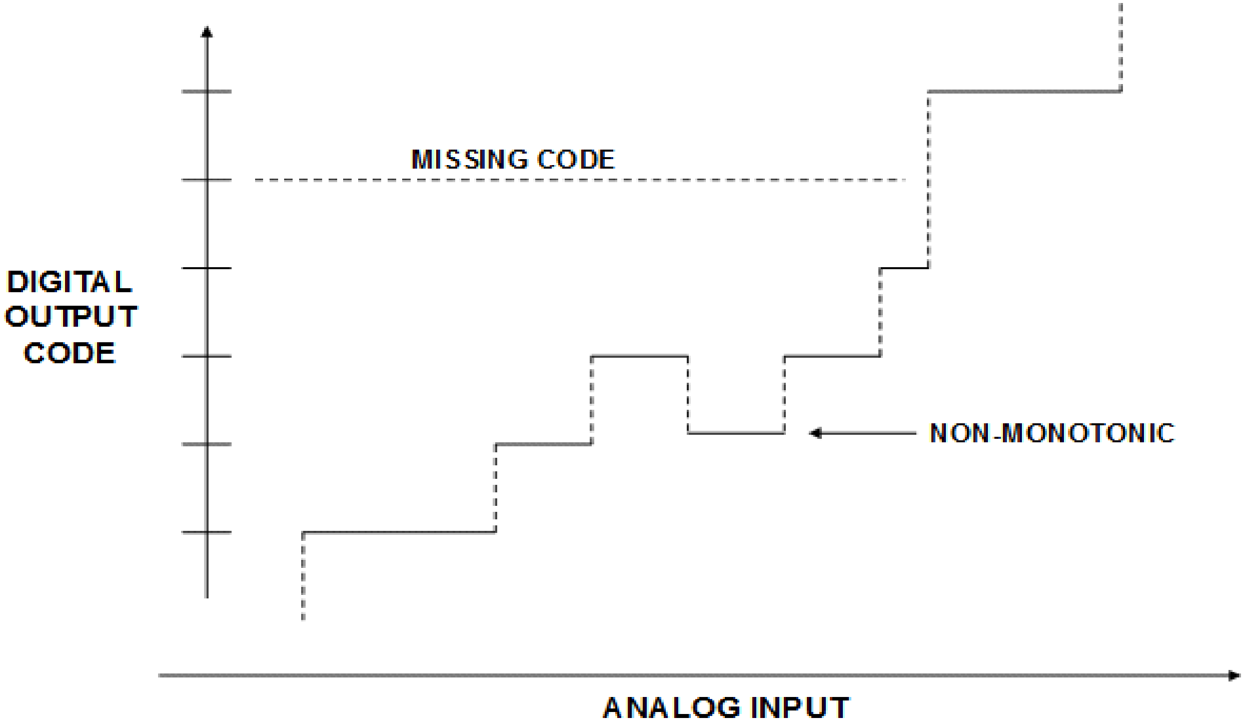
ANALOG INTERFACING

Differential non linearity error (DNL) for an ADC:



ANALOG INTERFACING

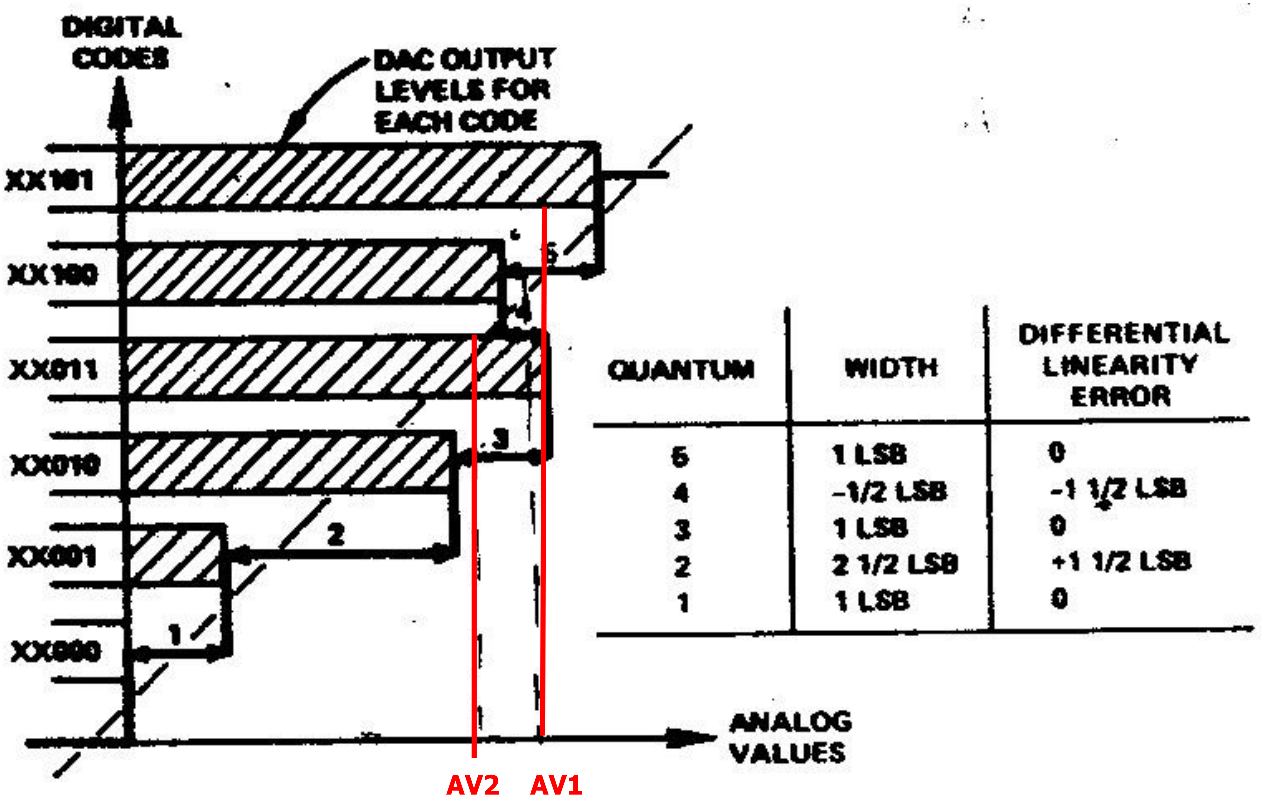
Differential non linearity error (DNL) for an ADC:



Non monotonic criterion for ADC

ANALOG INTERFACING

Differential non linearity error (DNL) for an ADC:



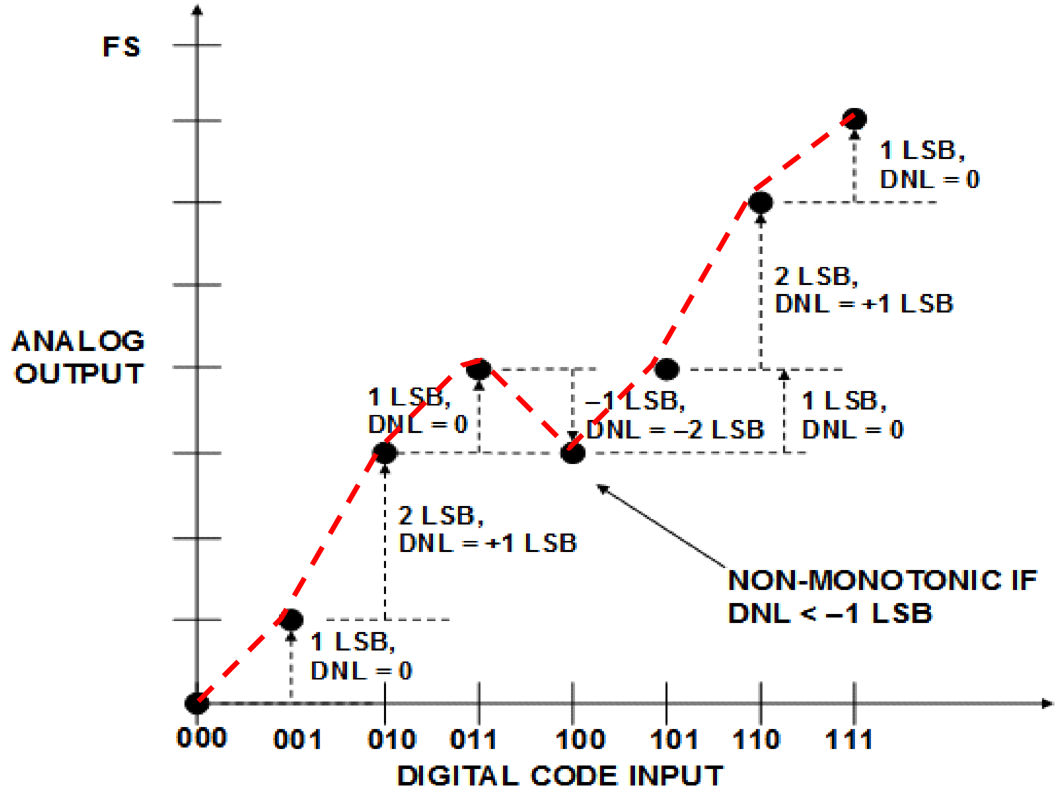
QUANTUM	WIDTH	DIFFERENTIAL LINEARITY ERROR
6	1 LSB	0
4	-1/2 LSB	-1 1/2 LSB
3	1 LSB	0
2	2 1/2 LSB	+1 1/2 LSB
1	1 LSB	0

Successive approximation ADC

Missing code

ANALOG INTERFACING

Differential non linearity error (DNL) for an DAC:



Non monotonic behaviour

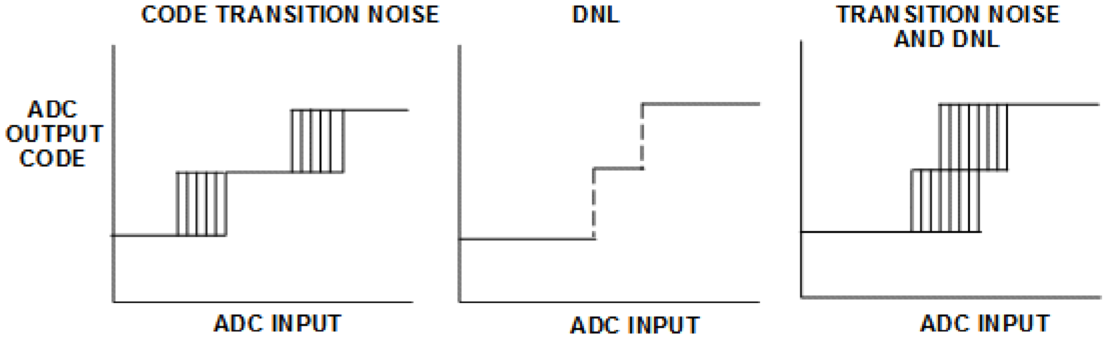
DNL < -1 LSB

ANALOG INTERFACING

Code transition noise + DNL for an ADC:

Code transition noise = error due to the noise in the electronic components of the ADC. This noise affects the real input signal and determines a output value that refers to a range > 1 LSB

The greater is the converter resolution the bigger is this effect



If the CTN is added to DNL, it can affect all the input values of a range

It can also happen that no inputs provide a specific output value although a range of input values that can be converted in that specific value.

This effect is present from 12 bit on.

ANALOG INTERFACING

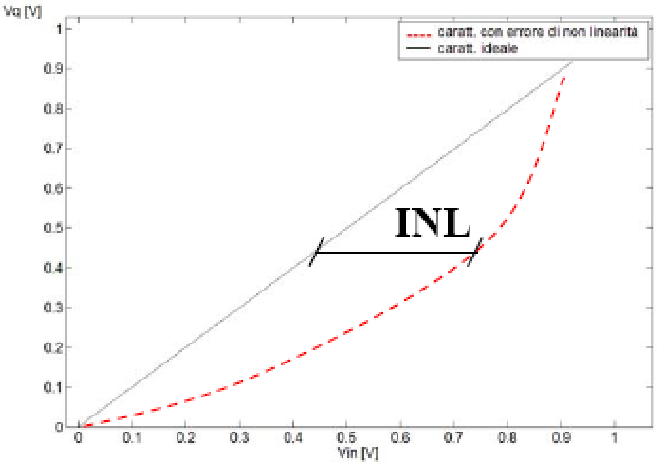
Integral non linearity error (INL):

DNL is a specific point parameter

The INL describes the maximum deviation of the real curve from ideal one so measuring the overall non linearity in the ADC

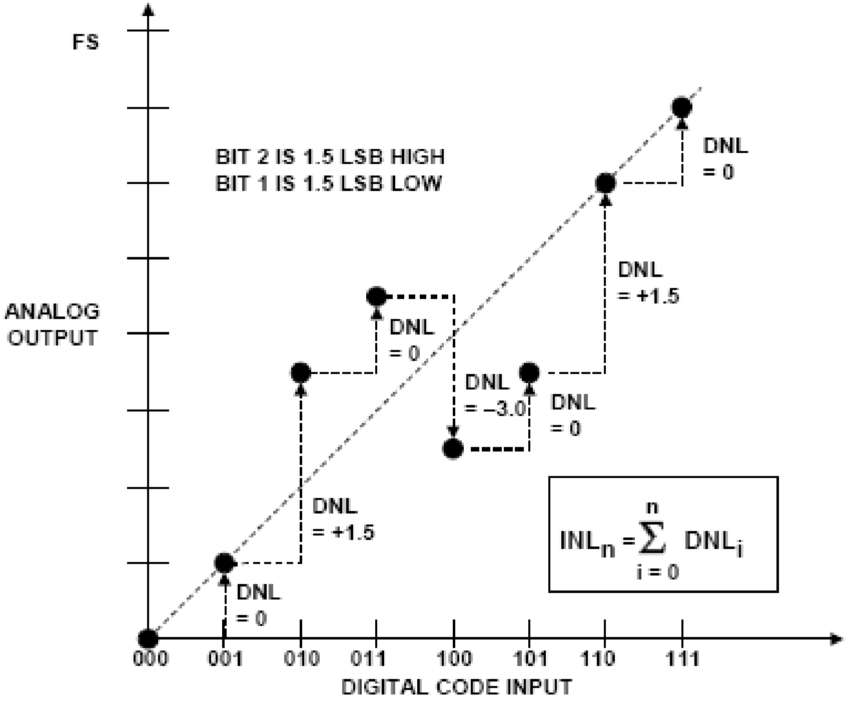
$$\mathrm{INL}_j = \sum_{i=1}^{j-1} \mathrm{DNL}_i$$

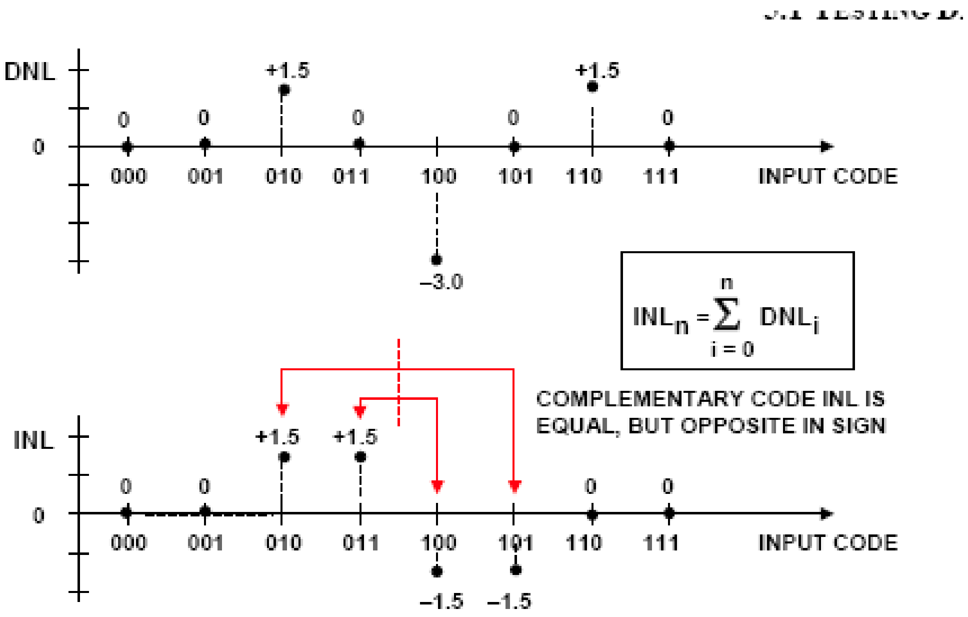
INL at the ith step



$$INL = \max_j (INL_j)$$

ANALOG INTERFACING





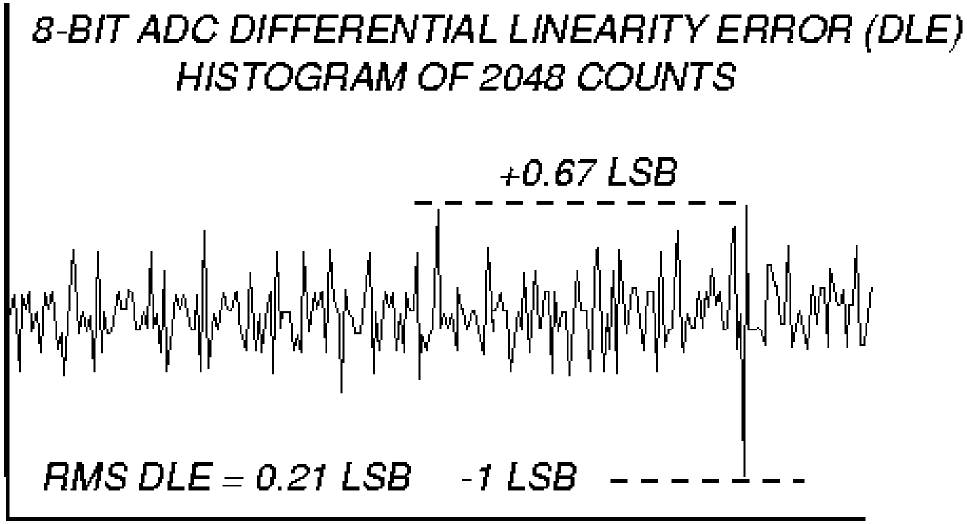
ANALOG INTERFACING

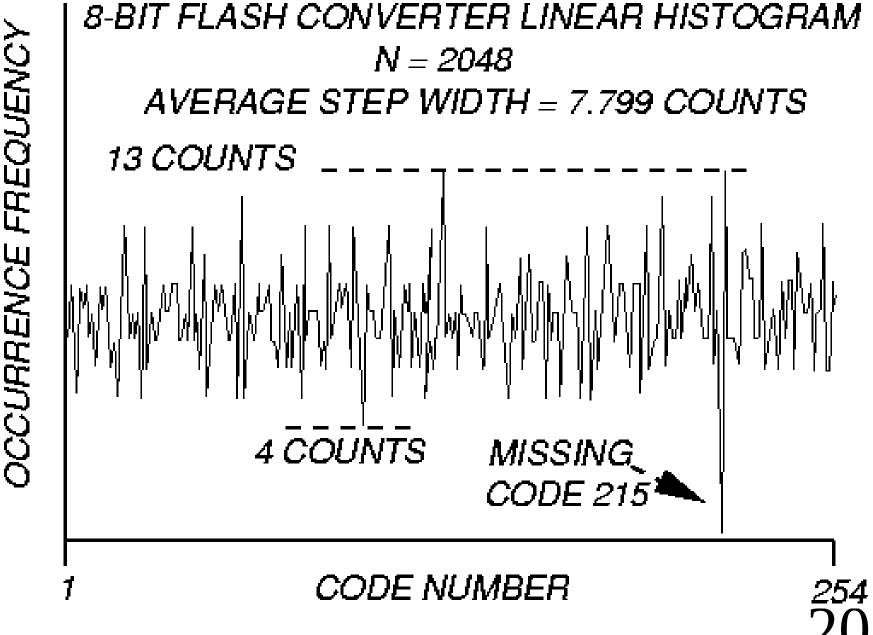
Non linearity error estimation:

A slowly variable signal given in input to the ADC (triangular or sinusoidal wave) with a small step between a value and the next one (i. e. 1 mV on 10V)

All the outputs are read and counted (frequency distribution of the output numbers)

Repeat the test and average: each count realtive to every output should ideally be V_{max}/(step*2^N). The deviation is an average evaluation of DNL

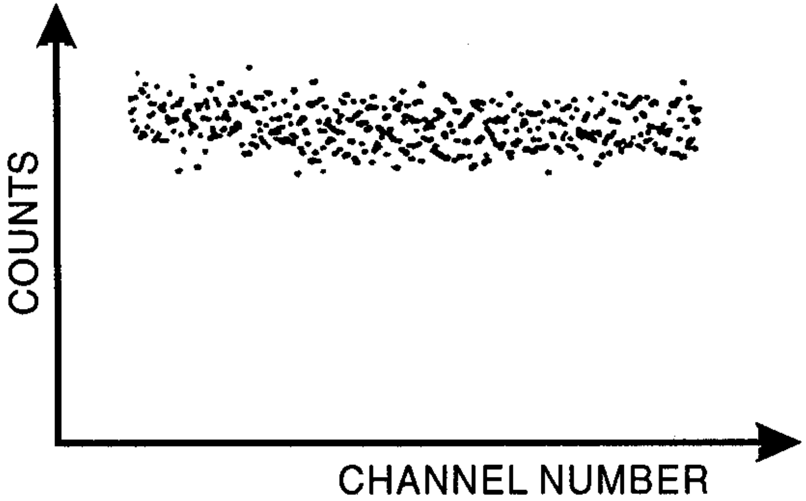


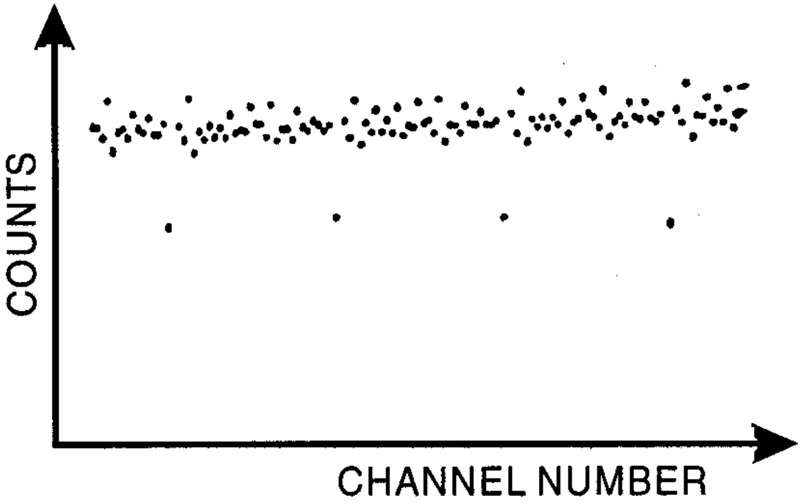


ANALOG INTERFACING

The count histogram should be ideally flat

The presence of a DNL error typically introduces periodic deviations





ANALOG INTERFACING

How to characterise ADC converters

Techniques that allows to estimate precisely and accurately offset, gain DNL and INL errors

Synthetic indexes that provide an overall evaluation of the converter behaviour:

-Signal to Noise ratio (SNR)

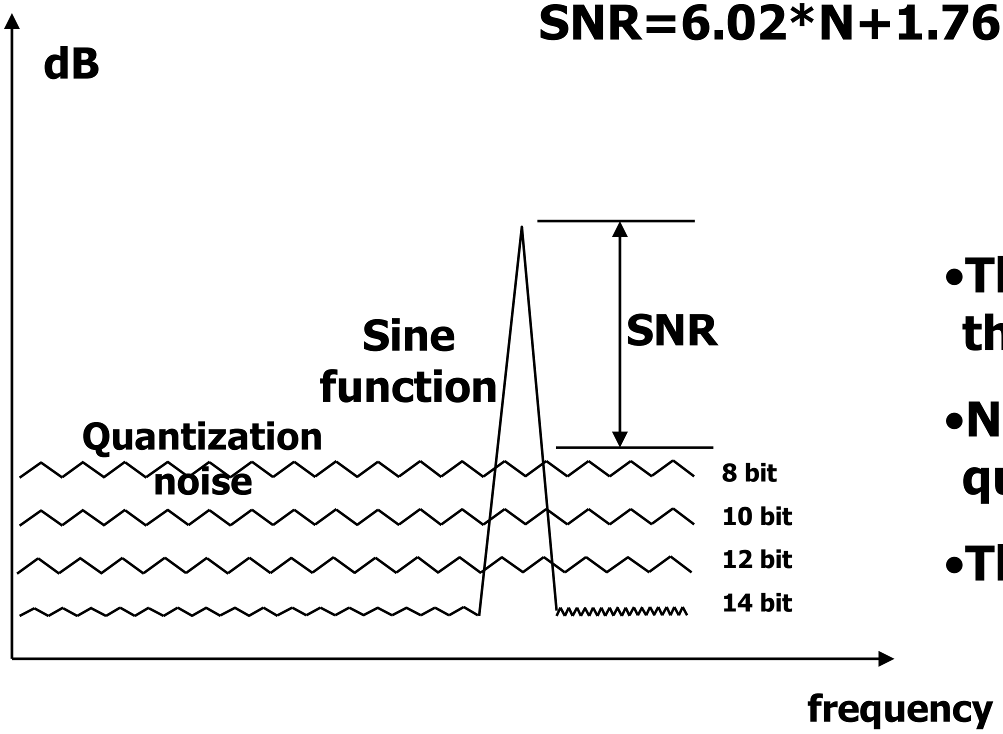
-Effective number of bits (ENOB)

ANALOG INTERFACING

Suppose that the quantization error is constant on all the range = Vref/2^N

We can demonstrate that (N number of bits):

SNR=6.02*N+1.76 (dB)



- **The SNR is evaluated through the FFT of a sine signal**
- **Noise power = ideal quantization noise**
- **The greater N the better SNR**

ANALOG INTERFACING

In the real case the SNR must include those inaccuracies due to non linearities, jitter … thus it will minor than in the ideal case.

From the previous expression we can say that:

$$SNR_{real}=6.02*N'+1.76\ (dB)$$

where N is the real number of bit of the used converter (ENOB), from which

$N' = ENOB = (SNR_{real} - 1.76)/6.02$

$SNR_{real} < SNR_{ideal} \Rightarrow ENOB < N$ **(ideal bit number)**

ANALOG INTERFACING

Flash converters:

N bit number

Resistive network with 2^N-1 comparators

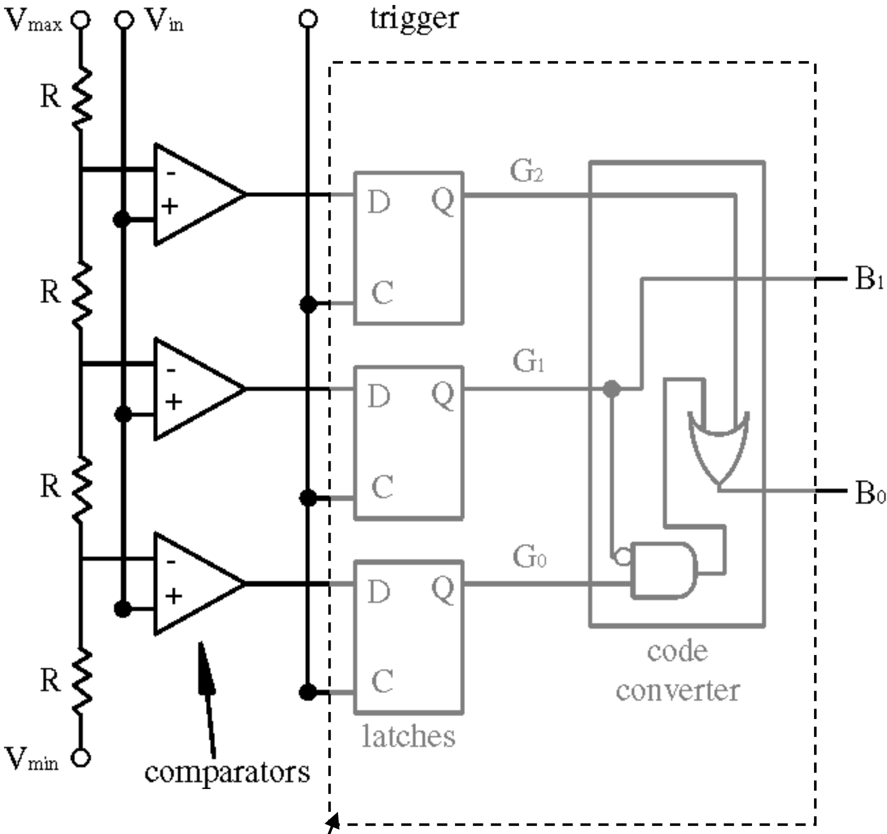
Low accuracy: 4-10 bit

Many bits would require complex circuits, with bigger encoders where it will be difficult to keep fixed those voltages values presented at the network nodes

High conversion velocity (10-100 nsec)

B_1=G_1

B_0=G_2 + G_0 * notG_1



Priority encoder

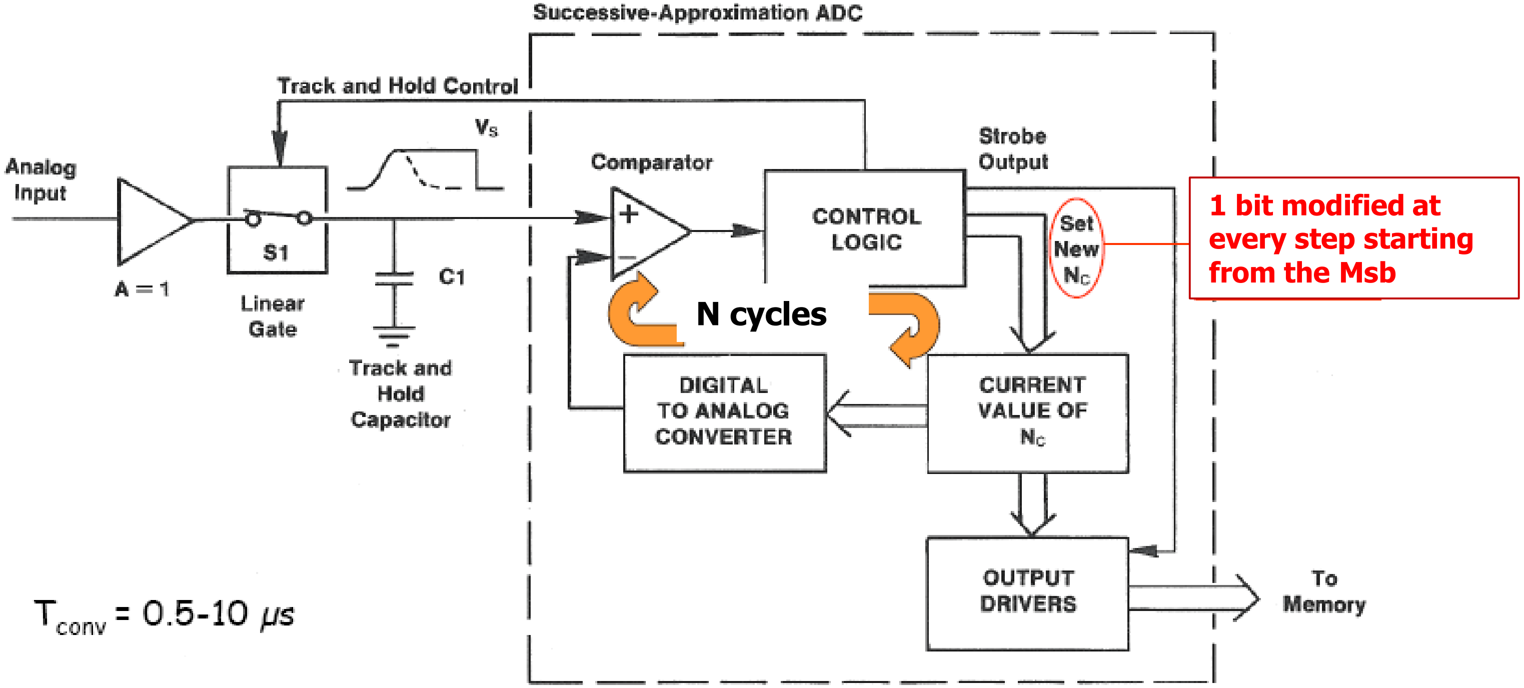
ANALOG INTERFACING

Successive approximation converter:

New value is proposed – DAC conversion – comparison – next value choice

From MSb to LSb. Fast ADC (order of tenths μs.). Sometimes S&H is needed

Sometimes the output is serial this provides galvanic decoupling and less output pins, but it requires a serial to parallel conversion at the acquisition level.



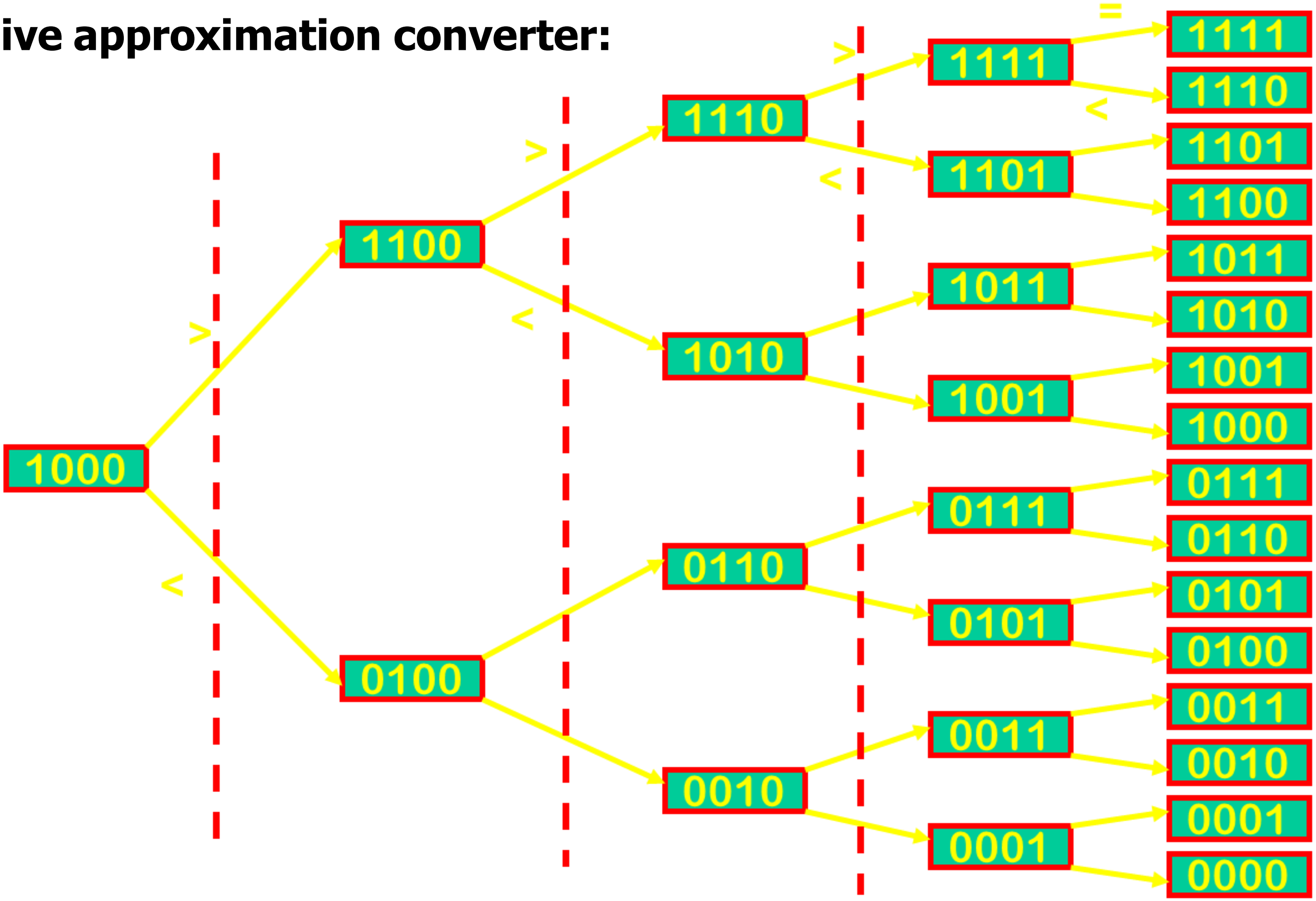
N cycles

1 bit modified at every step starting from the Msb

T_{conv} = 0.5-10 μs

ANALOG INTERFACING

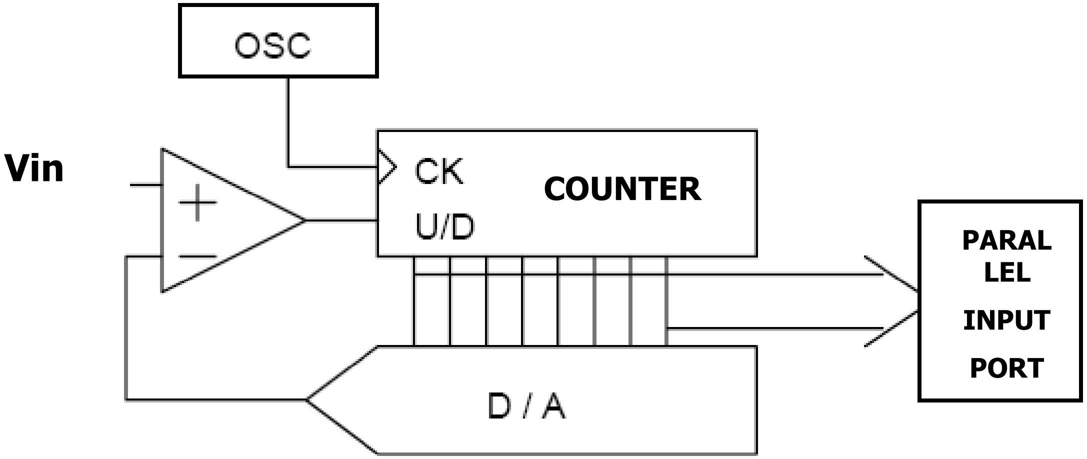
Successive approximation converter:



ANALOG INTERFACING

Tracking converters:

- **similar to successive approximation**
- **a free running up/down counter provides a value that is compared with V**
- **no multiplexer ⇒ the variable input values during commutations are followed (tracked)**
- **conversion time: 500 nsec – 10 μsec**



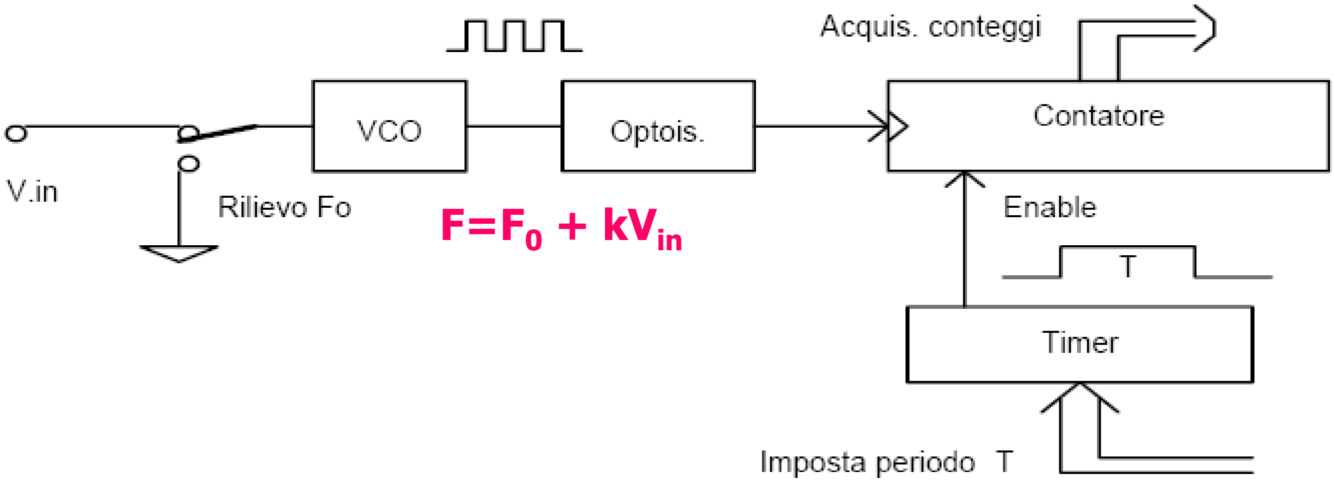
Max "slew rate" tolerated

$$\frac{dV}{dt} = \frac{V_{FS}/2^{N}}{T_{ck}}$$

ANALOG INTERFACING

Voltage frequency converter:

- **A Voltage Controlled Oscillator is used at a known frequency (MHz) who outputs pulses at a frequency proportional to the input voltage**
- **Pulses are counted within a established time interval**
- **The measurement is an "average" ⇒ white noise is rejected**
- **no S&H. Tradeoff between conversion velocity and accuracy**



$F=F_0 + kV_{in}$

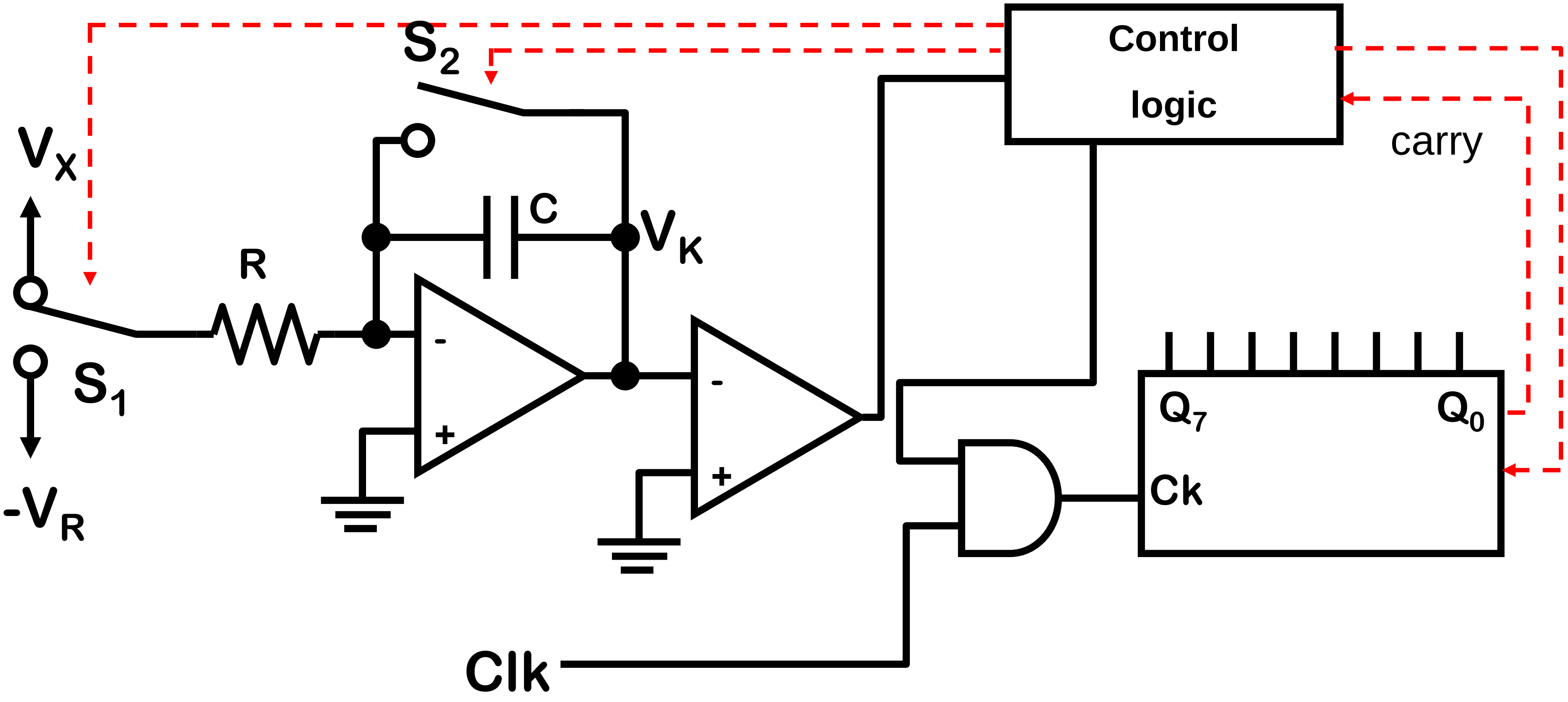
F_{VCO}=1 MHz, counting interval = 1 msec: 10 bit accuracy,

Counting interval 20 msec: 14-15 bit accuracy

ANALOG INTERFACING

Dual slope converters (Wilkinson):

The time for capacitor charging/discharging is measured through a counter

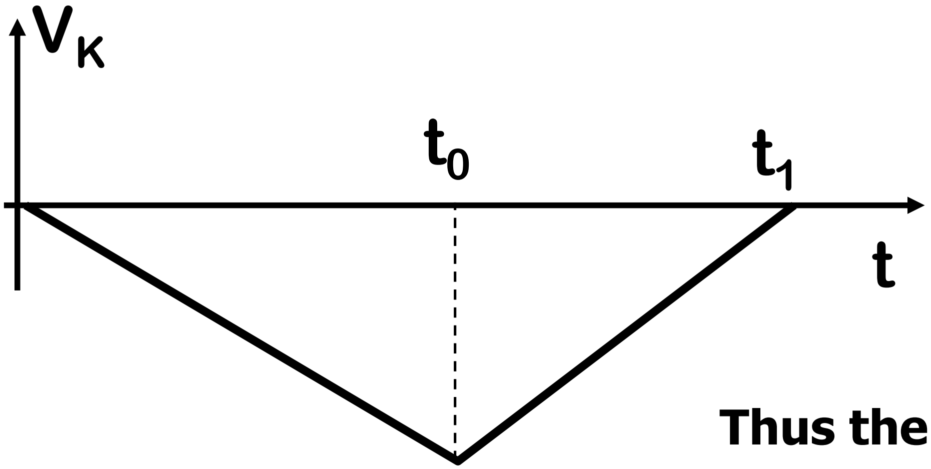


ANALOG INTERFACING

Dual slope converters:

At the beginning S_2 is closed, counter = 0. Then S_2 is opened and S_1 switched on V_x. The capacitor charge diminishes until the overflow of the counter in t_0; the counter sends a carry signal to the control logic. $V(t_0)=-V_x t_0/RC$

Now S_1 is switched on – V_R and the capacitor charges with velocity V_R/RC crossing zero in t_1 when the comparator sends another logic signal to the control logic. This allows us to say that:



$$t_1 - t_0 = \frac{V_x}{V_R} t_0$$

multiplying both the terms by F_{clk}

$$n_x = 2^N \frac{V_x}{V_R}$$

Thus the number n_x provided by the counter represents an estimation of the input voltage.

ANALOG INTERFACING

Dual slope converters: considerations

- **High resolution (16+ bit) and accuracy: the estimation is independent on electronic components that can change during their life so making the measurement less reliable**
- **no DNL**
- **Slow conversion: t_{conv} = 100-500 msec**
- **V_R, R, C e Fclk must be stable**
- **A display with 5/6 digits**
- **White noise and disturbs filtered by the capacitor**
- **The measurement is an "average"**

ANALOG INTERFACING

Time considerations:

The velocity of the chain components determines the times of activation of the control signals by the μprocessor

- **If multiplexing is used, $T_c=\Sigma T_{channel\_sampling}$**
- **If fly capacitor mux+ successive approximation ADC is used relay commutation times in the mux dominate (msec)**
- **If S&H is used, each mux channel can be switched during the ADC conversion**
- **If fast mux and ADC are used, the operational amplifier settling time dominates.**
- **Be careful to out of range input voltages that can saturate the amplifier. The exit from the saturation can be slow (100 μsec)**
- **If dual slope ADC is used ⇒ low f_c**

ANALOG INTERFACING

Sigma Delta converters (ΣΔ):

- **At the beginning used for audio transmission of the voice signals**
- **Then use was extended to high precision applications (24 bit)**
- **The idea was born in the first years of development of PCM (Pulse Code Modulation) but their use begins only with ‘70s**
- **Basic concepts: *delta modulation, noise shaping, oversampling and decimation***
- **Web reference in depth studies:**

Devices Analog to Digital Converters Tutorial MT-022

ADC Architectures IV Sigma-Delta ADC Advanced Concepts and Applications.mht

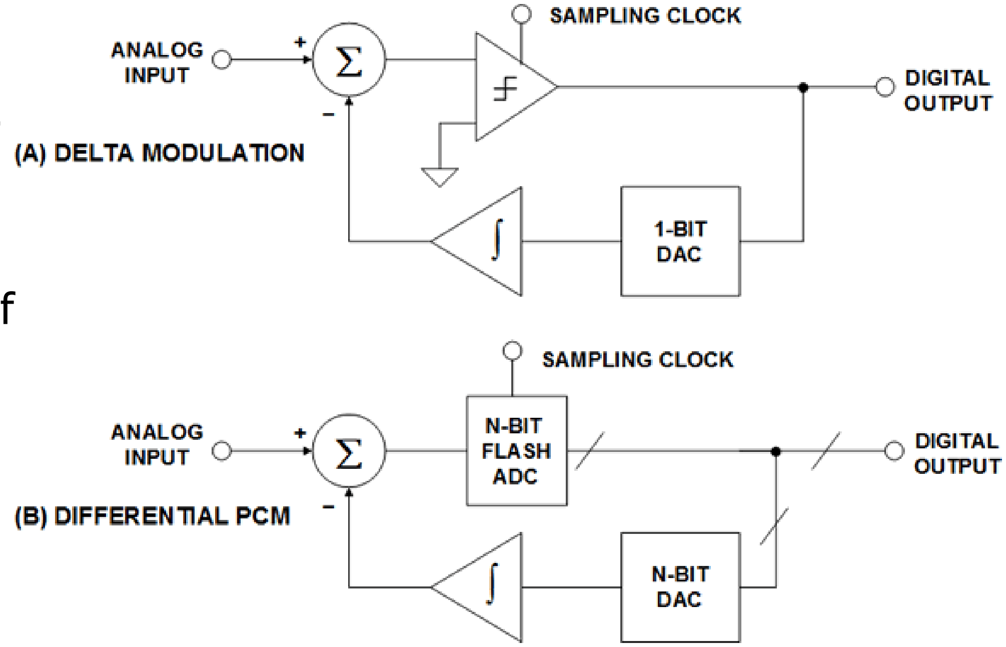
ANALOG INTERFACING

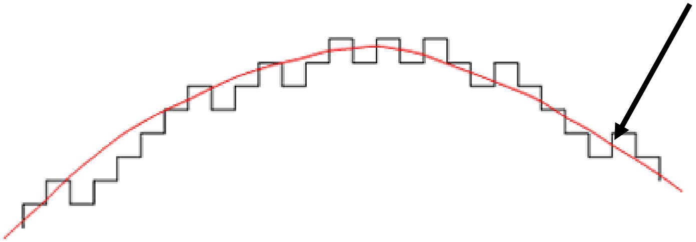
Delta Modulation and differential PCM:

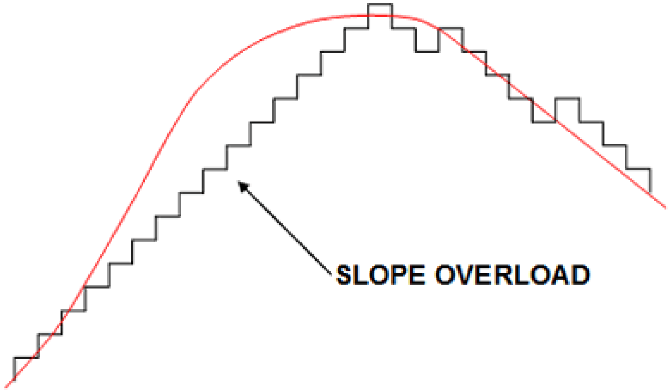
Delta Modulation:

A signal is converted in a sequence of zero's or one's by a comparator, converted in analog signal by a DAC and then integrated.

The analog signal is transmitted as sequence of 1 if a positive increment in the analog input occurred since the last time, 0 otherwise.







Differential Modulation:

Same idea but converters with more bits. We consider the first approach for sake of simplicity.

ANALOG INTERFACING

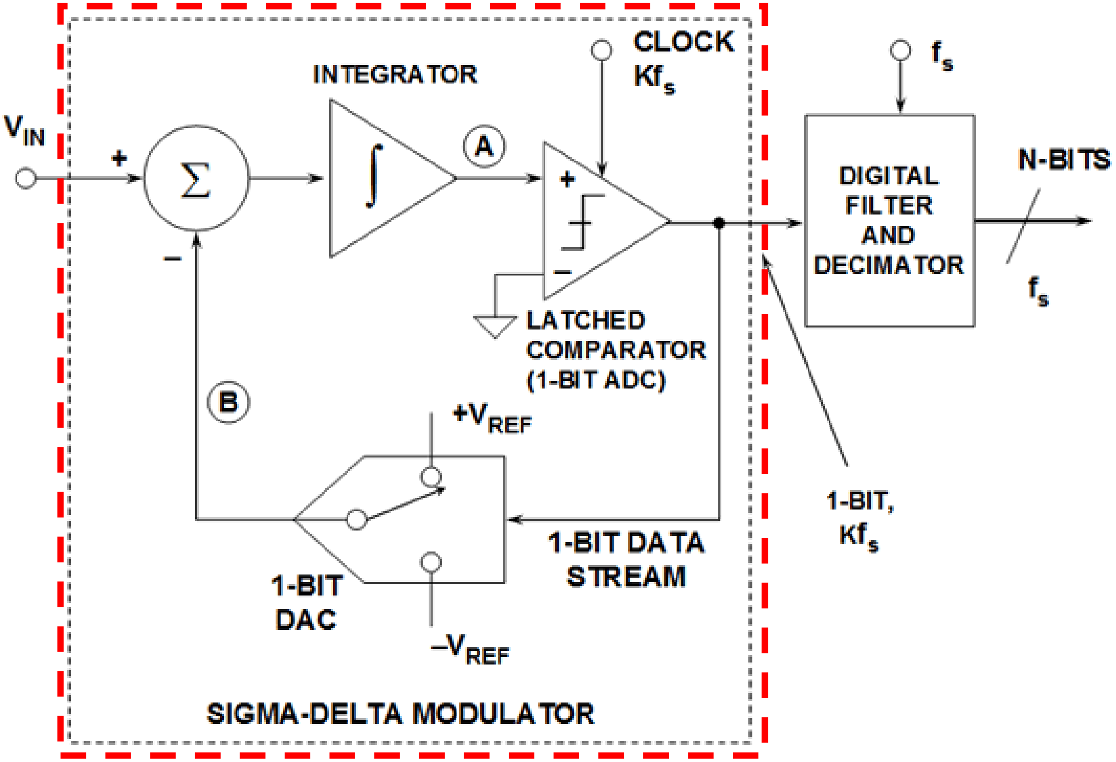
Sigma Delta modulation stage

If V_{IN} is continuous or stable (S&H) the integrator provides a triangular wave (A) that is compared with 0V through a comparator. Its output will be a train of “1s” e “0s” depending if the signal is ≥0 or <0.

This bit stream is basically a digital signal that drives the error between V_{IN} and the B input at the Σ block.

The average of this error (integral) reproduces the value of V_{IN}: to this purpose the result of the comparison drives a 1 bit DAC (basically a switch between two voltage levels +/- (B)).

I. e. the DAC output is compared with the input one and the error is averaged by the integrator that regulates the pulse emission.



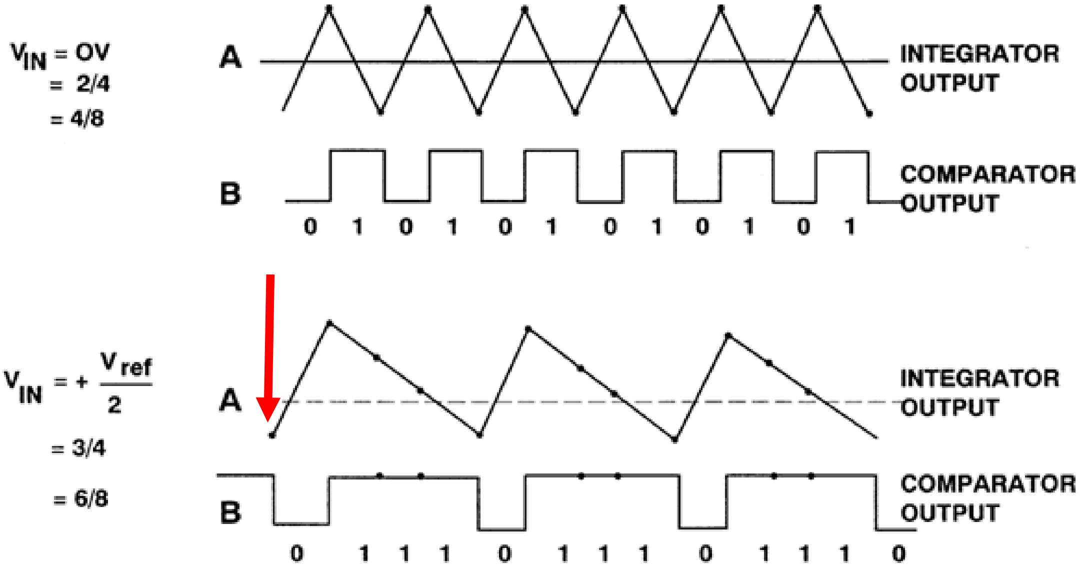
ANALOG INTERFACING

Example output of the integrator (similar to the voltage frequency ADC)

1) V_{IN}=0

The integrator provides a triangular wave centered on a constant value while the comparator produces a pulse train alternatively equal to “1” and “0”: if the filter samples at a frequency 4 times the signal one, the output of the comparator is 2/4 (2 out of 4 samples are high).

The bigger is the sample number considered (Kf_s) the better is the accuracy (i. e. with 8 samples, the output is 4/8 ⇒ 3 bit).



2) V_{IN}= V_{REF}/2

In this case the signal is mostly positive and the output is 3/4 if 4 samples are considered or 6/8 if 8 are used.

Example if V_{REF}=2 V and V_{IN}=1V (at the beginning A/B=0)

Σ =	**1,**	**-1,**	**-1,**	**3,**	**-1,**	**-1,**	**-1,**	**3**
A =	**1,**	**0,**	**-1,**	**2,**	**1,**	**0,**	**-1,**	**2,**
Cmp =	**1,**	**1,**	**0,**	**1,**	**1,**	**1,**	**0,**	**1,**
B =	**2,**	**2,**	**-2,**	**2,**	**2,**	**2**	**-2,**	**2**

ANALOG INTERFACING

Oversampling

A) The *q* quantization error in a ADC is $V_{REF}/2^N$.

The RMS of the quantization error in a sampled and converted signal is instead $q/(12^{1/2})$, uniformly distributed within 0 e f_s.

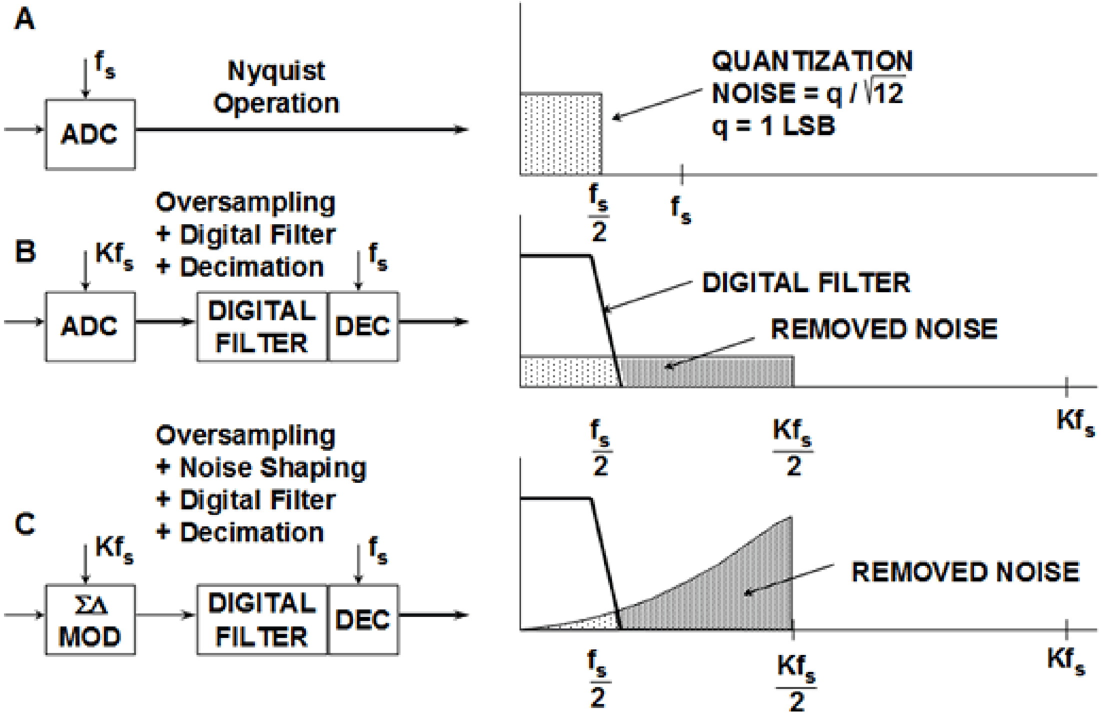
C) This happens in a normal ADC.

In a ΣΔ converter, however, the presence of an internal component of low pass type causes a reshaping of the noise (*noise shaping*) that will be mostly concentrated at high frequencies. This aspect further improves the SNR since the big part of the noise will be pushed at those frequencies that will be after cut off.

B) If we sample at a higher frequency (Kf_s) the quantization error remains similar but will be distributed on a wider frequencies range (0…Kf_s)

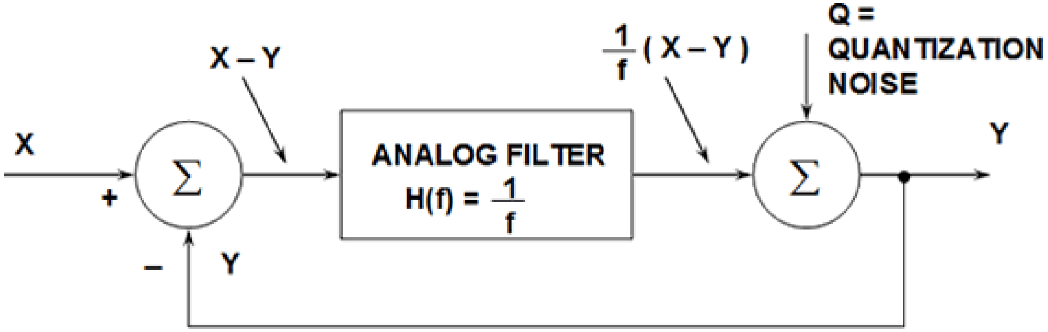
Now if we apply a low pass filtering, the filter removes the noise improving the ENOB.

K is the *oversampling* factor.



ANALOG INTERFACING

Noise shaping



If f goes to zero

$$Y=X$$

If f reaches infinite

$$Y=Q$$

The filter effect is low pass type on the signal and high pass type on the noise.

$$Y = \frac{1}{f}(X-Y) + Q$$

REARRANGING, SOLVING FOR Y:

$$Y = \underbrace{\frac{X}{f+1}}_{\text{SIGNAL TERM}} + \underbrace{\frac{Qf}{f+1}}_{\text{NOISE TERM}}$$

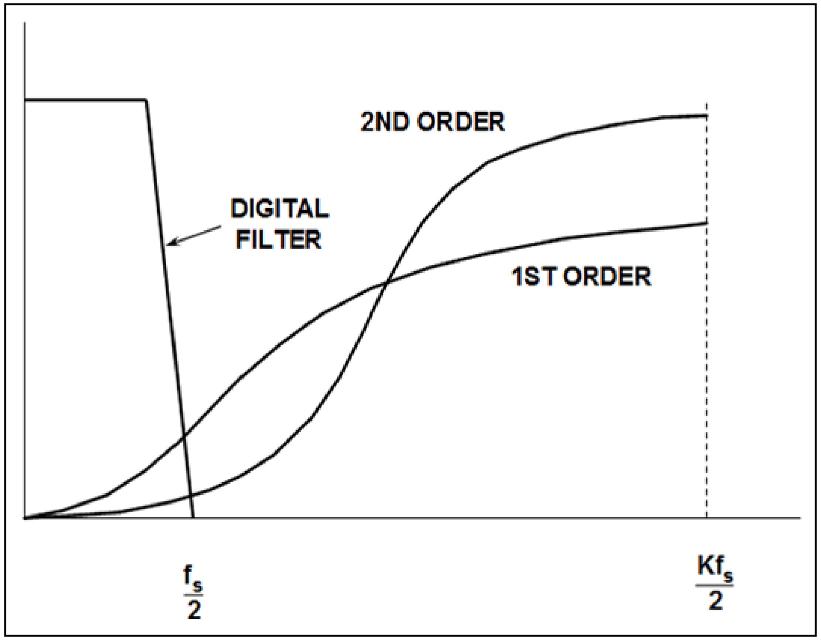
Decimation

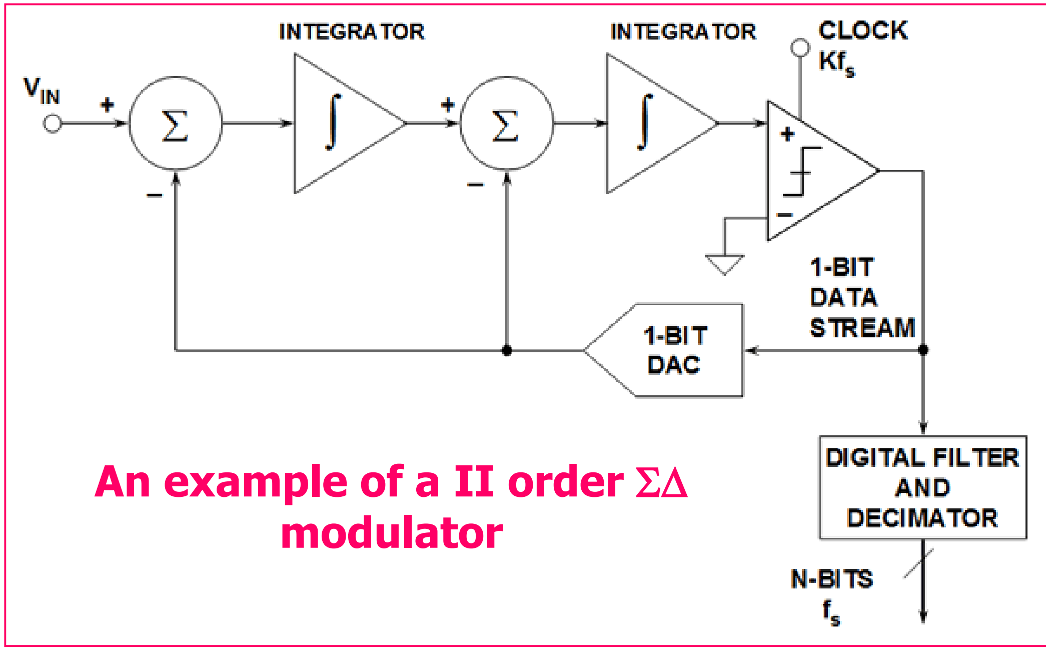
Since the bandwidth is reduced due to the filtering, the output is allowed to be at a minor frequency than the original one (Kf_s) but must still respect the Shannon theorem ($2f_s$). This can be obtained by giving in output only the M^{th} term while discarding the others.

This process is called “decimation” by an M factor.

ANALOG INTERFACING

Higher order ΣΔ modulators



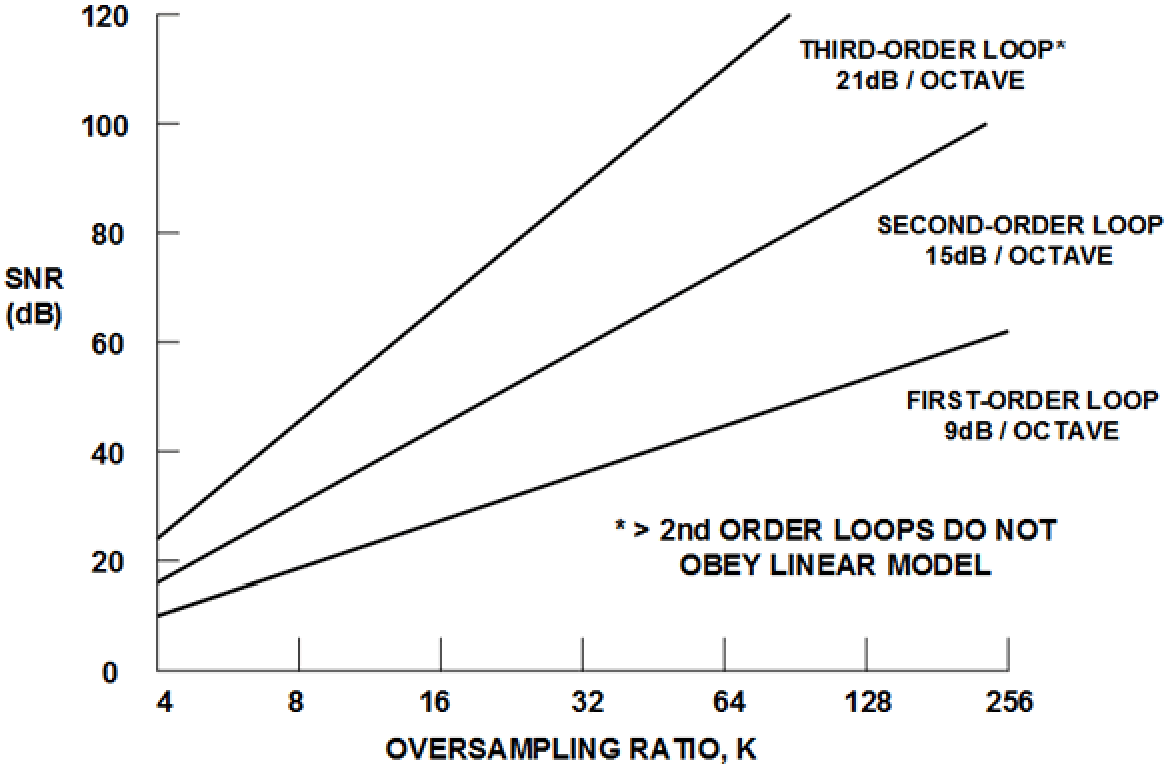


An example of a II order ΣΔ modulator

If we increase the number of integration and compare stages we obtain higher order ΣΔ modulators and generally speaking a better Noise Shaping and a better ENOB, if the oversampling is kept fixed

ANALOG INTERFACING

Project options



A suitable oversampling can be chosen to obtain a certain SNR, given a specific modulator.

If K=64, a II order ΣΔ converter provides a SNR=80 db that corresponds to a ENOB equal to 13.

ANALOG INTERFACING

3 levels characterize the activity of the μprocessor charged with the acquisition task:

- *Acquisition level*: managing interfaces and commands/control of the acquisition chain components
- *Measurement level*: pre-elaboration of the acquired values (the raw value acquired is transformed in a measurement)
- *Elaboration level*: the acquired values are elaborated (statistics, FFT, …)

Acquisition

- Fixed point data (without sign or complement 2) typically 16/32 bits coded
- When the ADC feature less bits than those used by the μP, a problem due to the sign propagation arises
- Sign propagation vs. alignment

ANALOG INTERFACING

Sign propagation:

- Those bits that must be added reproduce the value of the MSb of the ADC
- The values provided by the ADC will be concentrated at the extremes of the range
- **Alignment:**
- The MSb of the ADC becomes the MSb of the data bus
- Equal intervals separate the ADC output values

A suitable re-scaling will be necessary by the acquisition microprocessor sw

Example: 2 bits ADC, with 4 bits data bus

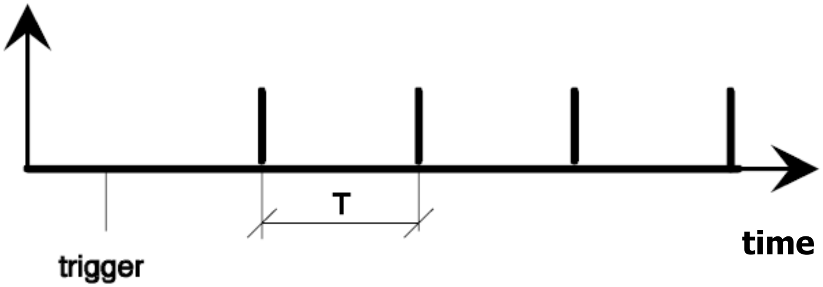
	propagation	alignment	direct (with loss of the sign)
00	0000	0000	0000
01	0001	0100	0001
10	1110	1000	0010
11	1111	1100	0011

ANALOG INTERFACING

Temporization

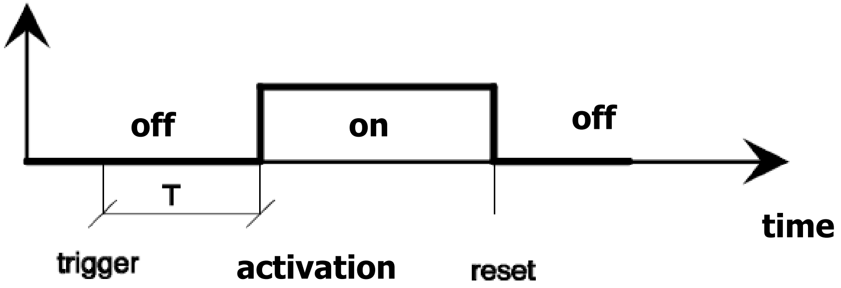
- ***Continuous cycle* to establish the sampling frequency (*wait*)**

  Time base = Real Time Clock if enough for the desired f_c (tenths msec, ok if slow components, ko if fast components)



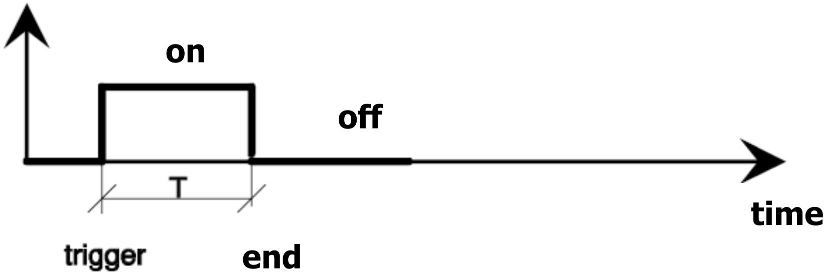
- ***With delay* to wait the settling of the signals used to trigger the chain components *(delay)***

  Order of 10-100 μsec, no RTC instead SW cycles or HW timers



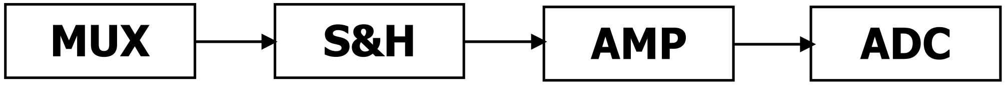
- ***Time stamping (read_timer)***

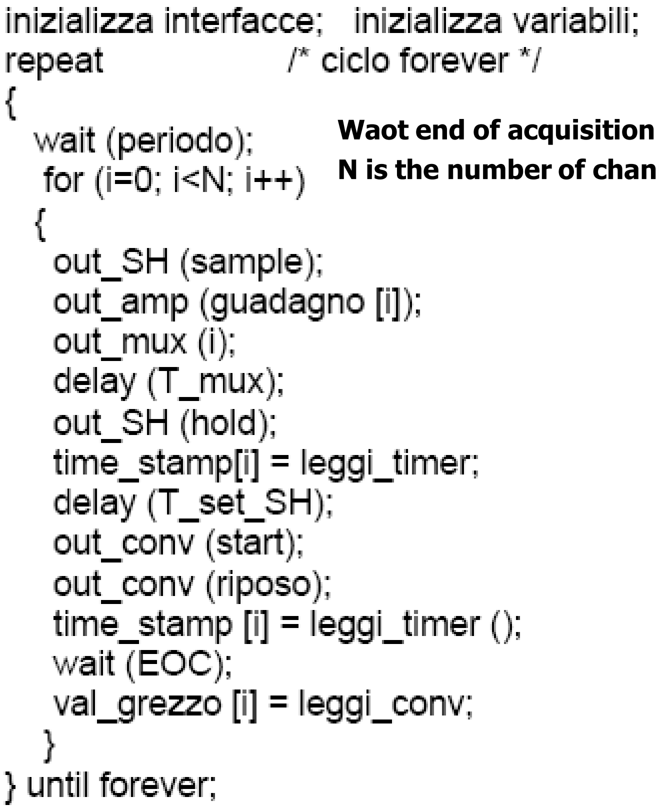
  Time corresponding to a sampling or another function. Use a free-running timer with suitable duration (10 μsec)



ANALOG INTERFACING

Example of a acquisition cycle:





```
inizializza interfacce;   inizializza variabili;
repeat                /* ciclo forever */
{
  wait (periodo);
   for (i=0; i<N; i++)
  {
    out_SH (sample);
    out_amp (guadagno [i]);
    out_mux (i);
    delay (T_mux);
    out_SH (hold);
    time_stamp[i] = leggi_timer;
    delay (T_set_SH);
    out_conv (start);
    out_conv (riposo);
    time_stamp [i] = leggi_timer ();
    wait (EOC);
    val_grezzo [i] = leggi_conv;
   }
} until forever;
```

Waot end of acquisition period (sampling frequency)

N is the number of channels

Open S&H
Gain set
Channel selection
Wait for commutation
Hold S&H
Time_stamping
Settling time S&H
Start of conversion
End Start of conversion
Read what time is it
Wait End of conversion
Read data

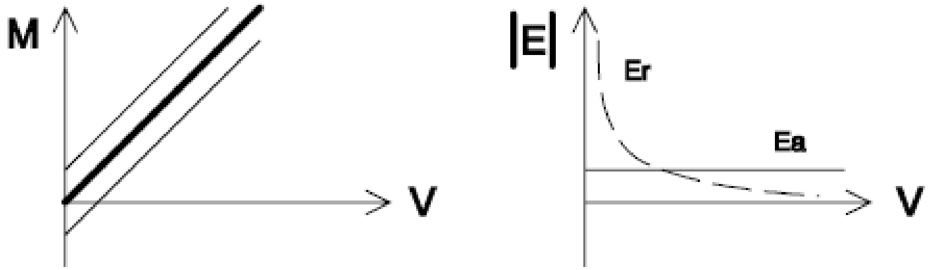
Same sampling frequency for every channel

ANALOG INTERFACING

An adjustable (programmable) gain in Op Amp

CONSTANT ABSOLUTE ERROR (Ea)

The error range of the measurements is constant when the measured variable V changes. This can be typically be caused by disturbs independent on the value of V, by an offset in the measurement instrumentation or by a fixed point binary representation of the numbers.



The two graphs show the behavior of the absolute error Ea and the relative error Er compared to the measured variable V (M is the value of the measurement).

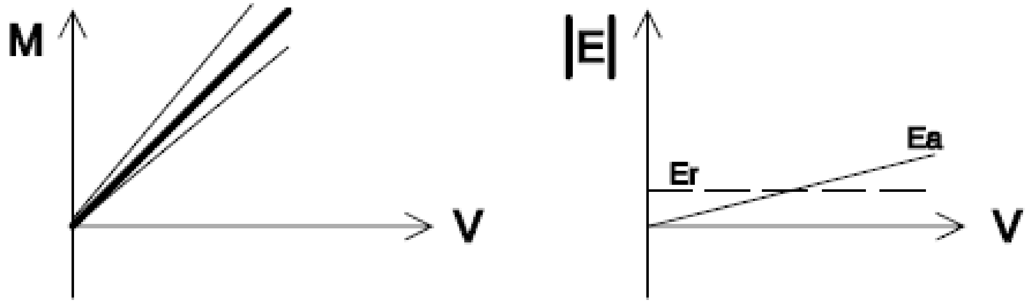
$$E_a = M\text{-}V \qquad E_r = (M\text{-}V)/V \sim E_a/M$$

ANALOG INTERFACING

An adjustable (programmable) gain in Op Amp

CONSTANT RELATIVE Er ERROR

Typically due to gain errors in the instrumentation or a floating point numbers representation.



The two graphs show the behavior of the absolute error Ea and the relative error Er compared to the measured variable V (M is the value of the measurement).

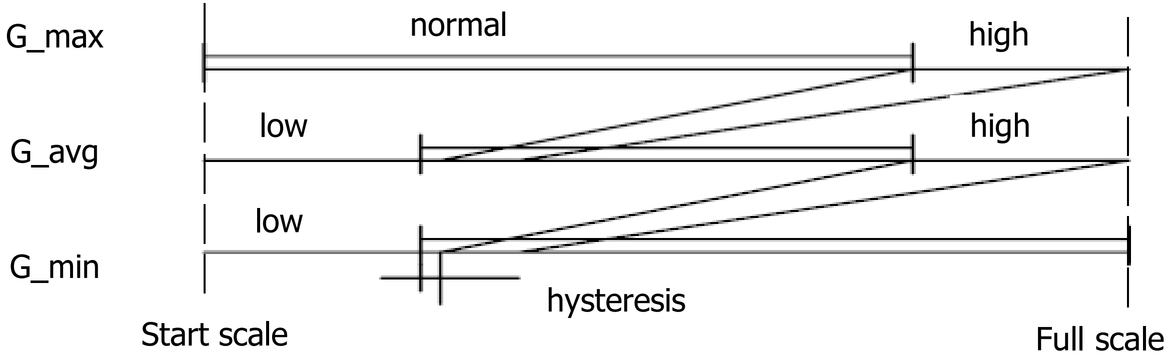
ANALOG INTERFACING

An adjustable (programmable) gain in Op Amp

- If a small quantization absolute error is required when the magnitude to be measured is small (E_r ~ constant) a logarithmic amplifier can be employed.
- As an alternative a programmable gain amplifier can be used so as to adapt the amplification to the range of values to which the input signal belongs.
- The amplification allows to provide the ADC with an input value that is in high part of its scale so as to reduce the influence of the quantization error.
- It is useful when the input signal features wide variations and the cost of the variable gain amplifier is acceptable.
- Another requirement is a slow signal compared to sampling frequency
- *Autorange*

ANALOG INTERFACING

An adjustable (programmable) gain in Op Amp



Different thresholds for different gains

Every time a new value is acquired (measured):

```
if measurement < low_threshold
    new_gain = higher gain compared to old_gain
else if measurement > high_threshold
    new_gain = lower gain compared to old_gain
else new_gain = old_gain
```

- Values are subdivided in as many intervals as the possible gains are.
- Each range is divided in 3 categories: “normal”, “high”, “low”.
- An interval and the corresponding gain is considered as “normal” if the quantization error is the range of the required one.
- Hysteresis area to avoid continuous commutations.
- The sw must register the set gain and adopt the corresponding conversion curve (measurement).

ANALOG INTERFACING

Measurement level

- The measurement is produced
- The datum given by the sensor is converted in measurement units (components gain, transfer function, …)
- Single floating point precision sometimes also double
- Required pre-processing:
    - Linearization
    - ADC conversion (16/32/64 bits)
    - Calibration
    - Derived measurement calculation
    - Digital filtering

ANALOG INTERFACING

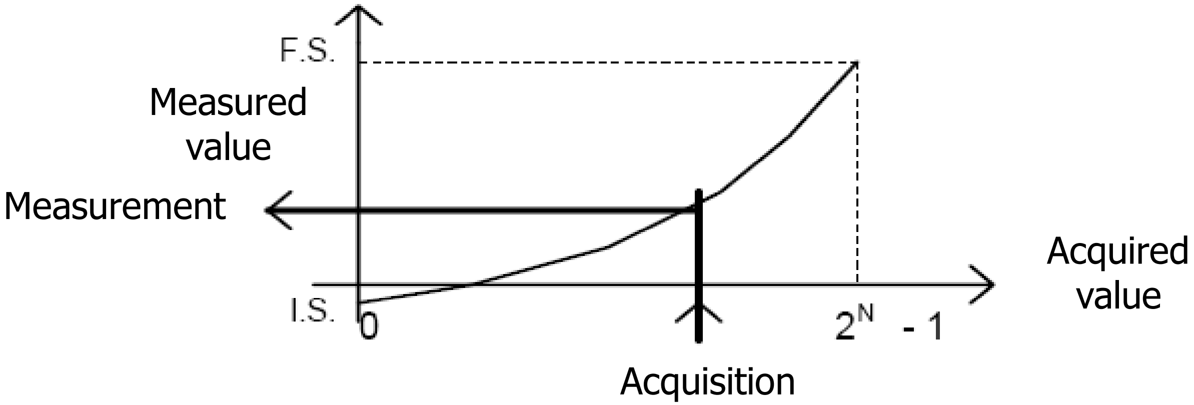
Linearization and conversion

The relationship between the acquired value (A) from the transducer and the measured one (M) (*characteristic curve*) can be represented by a line crossing or not zero but also by a generic curve

This characteristic is the product of all the characteristic of the chain components.

$0<L<2^N-1$ (if unipolar ADC) $-2^{N-1}<L<2^{N-1}-1$ (if bipolar ADC)

M is included within a *starting* scale and a *full* scale value.



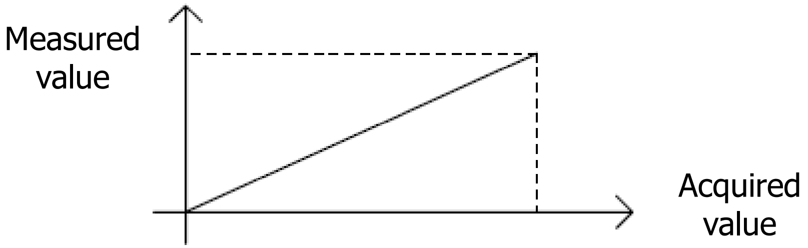
Measured value

Measurement

Acquired value

Acquisition

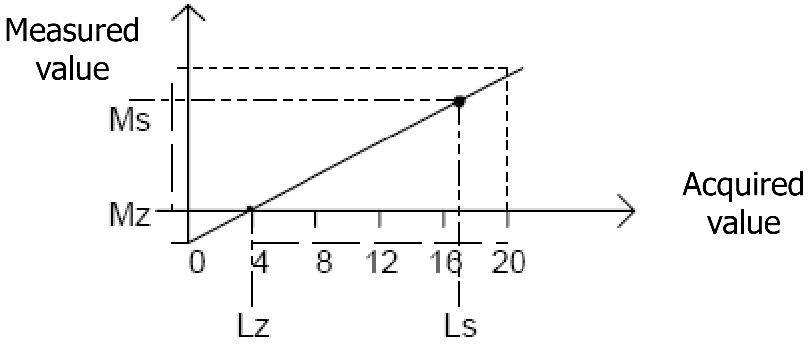
ANALOG INTERFACING



1) Zero crossing line

$$M=K_m*L$$

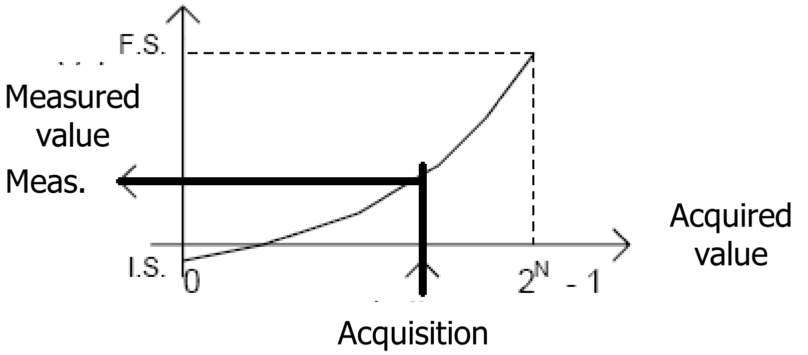
K_m is the inverse of the overall chain gain



2) Non zero crossing line

Two points are necessary to define the line: zero and span points (span = 3/4 full scale)

$$M=M_z+(L-L_z)(M_s-M_z)/(L_s-L_z)$$



3) Generic curve

- An array of points to be interpolated

- Analytic representation $M=AL^2+BL+C$

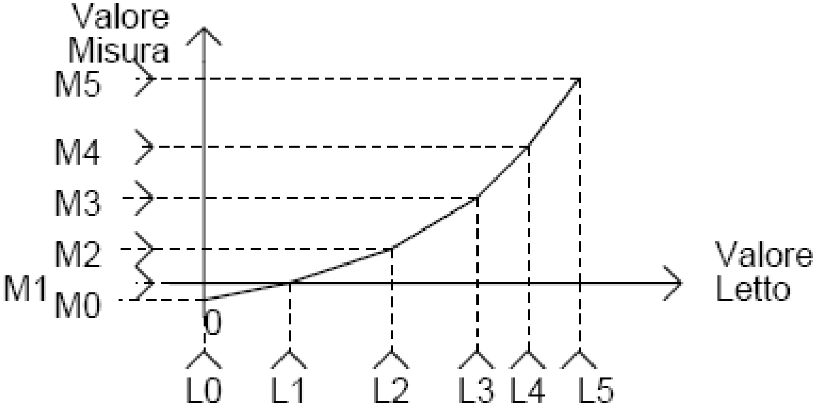
ANALOG INTERFACING

Calibration of the characteristic curve

Those parameters that describe the M-L curve may be not completely known or changing during the time due to thermal effects or usage.

Calibration = physical setting of the chain components so as each one works in nominal conditions (offset, gains, …). The final curve is the product of the nominal component characteristics.

Software calibration = “real” parameters acquisition, after the transient phases $\hat{L}_i$, by giving the system input known values of the measured magnitude (M_i) so as to determine the deviation from the previous curve. It must be done at the beginning and periodically.



Automatic or semi-automatic calibration (values provided by the PC or by the operator)

Finally parameters estimation through “best fitting” if the curve is generic, or through an array (M_i, L_i) if the curve is a $\hat{\square}_i$ line.

ANALOG INTERFACING

Derived measurements

Secondary magnitudes calculated from those acquired through a transducer (i. e. volumetric or mass flow rate, absolute temperature through a thermocouple, …).

If these values are obtained though a subtraction pay attention to the *quantization* errors.

If these values are obtained though an integration pay attention to *systematic* errors.

If measurements are obtained through derivation pay attention to *random* errors or *jitters* on sampling period since their can make the calculation to diverge.

If fast variable magnitudes are acquired ⇒ high sampling frequency.

ANALOG INTERFACING

Numeric filtering

The acquired values are replaced with more probable values obtained by deleting errors (considered as random) on the basis of:

- Physical model of the phenomenon
- Characteristic curve of the overall acquisition chain
- Model of the noise
- Model of the user behavior
- Used sampling frequency and necessary frequency

The numeric filtering allows:

- Good and stable filtering even with low cutoff frequencies
- Easy calibration of the characteristic parameters
- Better resolution of the used ADC

ANALOG INTERFACING

General expression of an ARMA filter

I(k) = kth input sample to the filter (present sample)

U(k) = value of the filter output at the kth instant

$$U(k) = -a_1*U(k-1) - \dots - a_n*U(k-n) + b_0*I(k) + \dots + b_m*I(k-m)$$

If a_i=0 **Moving Average** filter with finite answer to an input pulse

If a_i not 0 **Auto-Regressive** filter with infinite answer to an input pulse

Causal filters: the kth output is function of the (k-i) previous values. It is a direct (on line) filtering with a “delay effect”

Not causal filters: the kth output is function both of previous and of the successive values. It is a typical offline filtering (i. e. spatial image filtering)

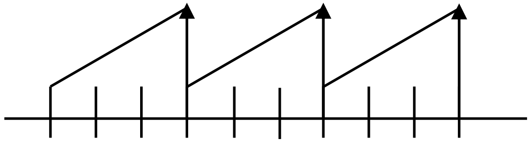
ANALOG INTERFACING

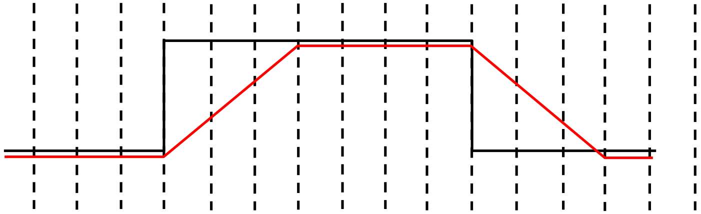
Moving average filters:

They differ from the “usual” *fixed average* ones that evaluate the average of the input signal on periods multiple of the sampling period and replace the last N samples with a sole value equal to their average

They are equivalent to an ARMA filter with a_i=0 and b_i=1/m

However each sample is replaced with the average of the last *m* samples





Fixed average filtering

Moving average filtering **U(k)=[I(k)+I(k-1)+I(k-2)]/3**

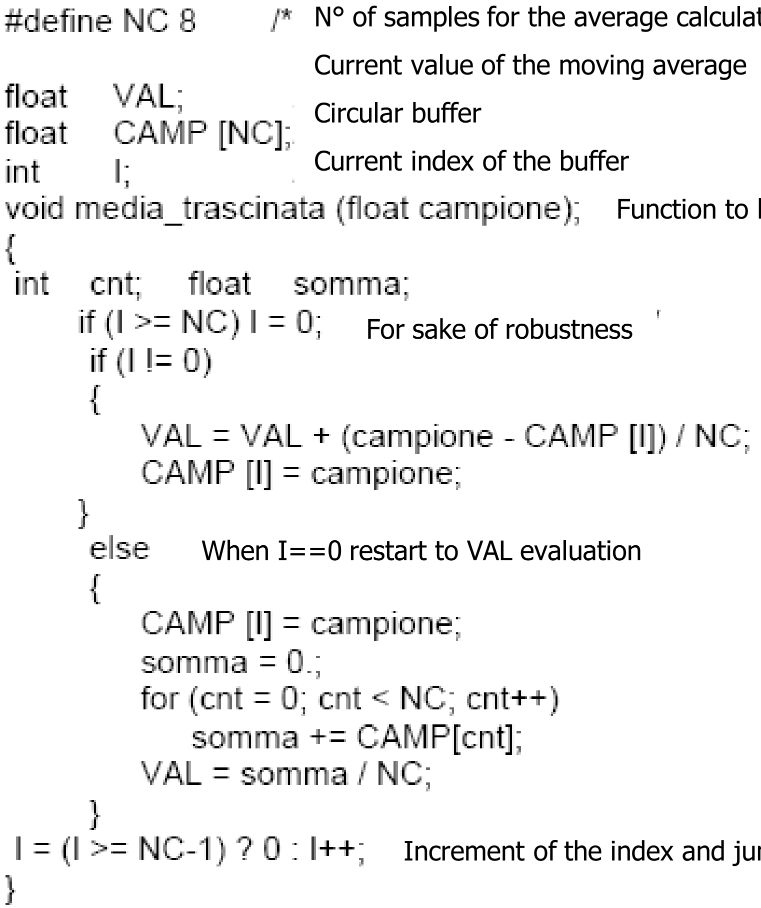
The answer to a square wave input is 0 after a time = mT (T sampling period).

Delay effect = T(1+m/2)

A memory is required to store the last m samples (circular buffer if m is high)

ANALOG INTERFACING

Example of moving average filter with m=8



```
#define NC 8        /*  N° of samples for the average calculation
                        Current value of the moving average
float    VAL;
                        Circular buffer
float    CAMP [NC];
                        Current index of the buffer
int      I;
void media_trascinata (float campione);    Function to be called at every acquisition to upgrade VAL
{
 int    cnt;    float    somma;
       if (I >= NC) I = 0;    For sake of robustness
        if (I != 0)
        {
            VAL = VAL + (campione - CAMP [I]) / NC;
            CAMP [I] = campione;
       }
        else    When I==0 restart to VAL evaluation
        {
            CAMP [I] = campione;
            somma = 0.;
            for (cnt = 0; cnt < NC; cnt++)
                somma += CAMP[cnt];
            VAL = somma / NC;
        }
 I = (I >= NC-1) ? 0 : I++;    Increment of the index and jump or exit
}
```

ANALOG INTERFACING

Exponential filtering:

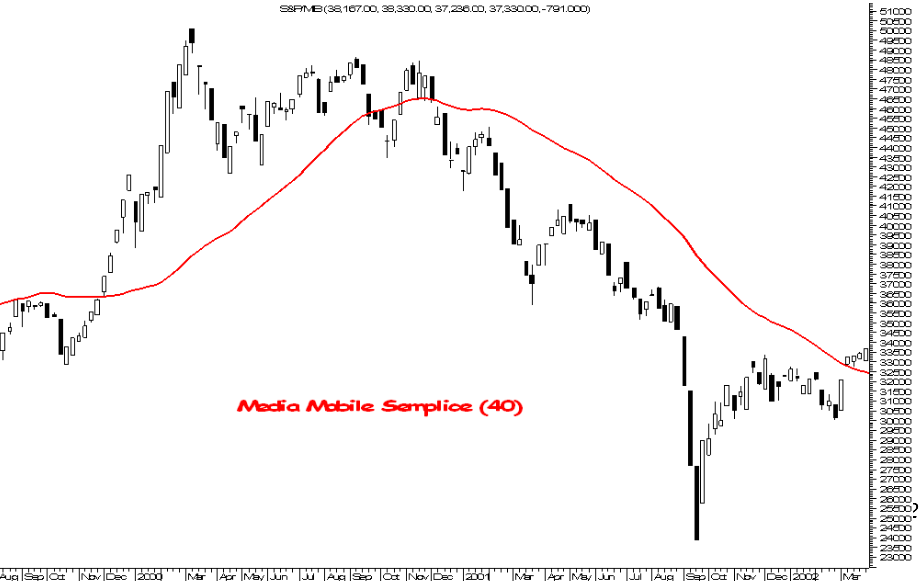
It’s an ARMA filter with a_1=-CF e b_0=(1-CF) and 0<CF<1

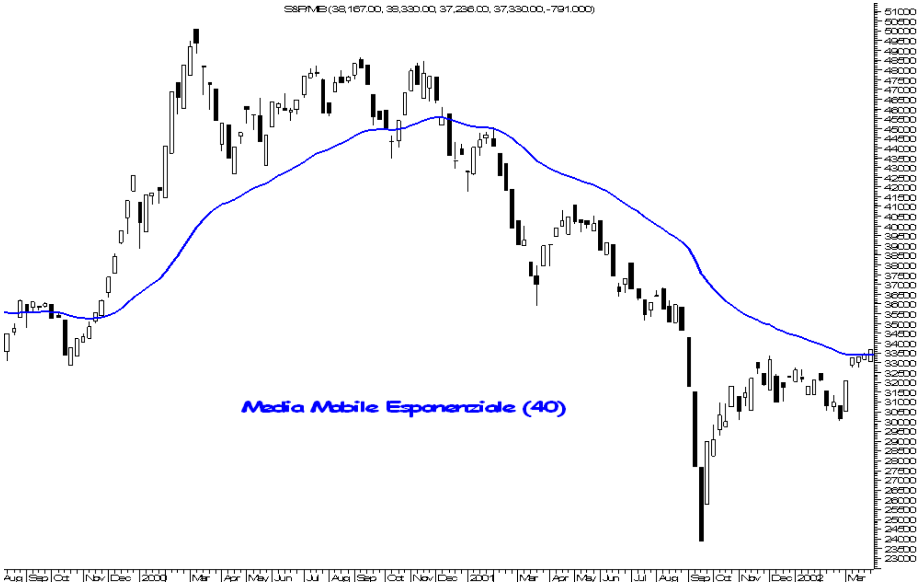
If CF low ⇒ the output is mainly due to the input effect

If CF high ⇒ the output is mainly due to its proper “history”

Time constant of the filter (a measure of its velocity) = T/(1-CF)

$$\mathbf{U(k) = I(k) + CF*[U(k-1)-I(k)]}$$





ANALOG INTERFACING

Example of an Intel assembly code implementing an exponential filter:

First order filtering working on fixed point data to speedup calculation

UF is the new filtered value (filter output) upgraded at every call

IR is the raw sample (filter input)

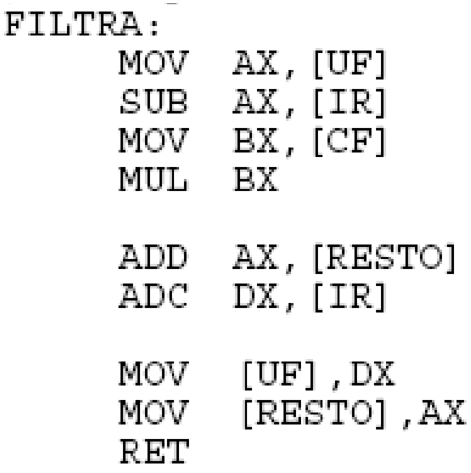
CF is the filter coefficient < 1 , expressed as a fraction of 2^{16} ; for example 0.5 ⇒ a CF = 32768 = 8000H

RESTO stores the fraction part of the result (remainder) ; this value must be recovered when the function is next time invoked

CF=CF*2^{16}

CF fixed point

CF*(U-I) in DX & AX



```
FILTRA:
     MOV   AX,[UF]
     SUB   AX,[IR]
     MOV   BX,[CF]
     MUL   BX

     ADD   AX,[RESTO]
     ADC   DX,[IR]

     MOV   [UF],DX
     MOV   [RESTO],AX
     RET
```

AX=U(k-1)-I(k)

The result is in DX,AX registers

Consider the previous remainder

Add the input with carry

Upgrade the filtered value

Save the new remainder

ANALOG INTERFACING

A numeric example:

if UF=25, IR=20, RESTO=0.25; CF=0.5=32768

```
FILTRA:MOV AX, [UF]
        SUB AX, [IR]            ;AX=5
        MOV BX, [CF]            ;BX=32768
        MUL BX                  ;DX=2, AX=32768
        ADD AX, [RESTO]         ;AX=2^15+2^14
        ADC DX, [IR]            ;DX=22
        MOV [UF], DX            ;
        MOV [RESTO],AX          ;salva nuovo resto
        RET
```

DX	AX	
0000000000000000	0000000000000101	
0000000000000010	1000000000000000	
0000000000000010	1100000000000000	
0000000000010110	1100000000000000	22,75

The expected result from the formula is U(k)=I(k)+CF*[U(k-1)-I(k)] = **20+0,5[25,25-20]=22,625**

Good approximation

ANALOG INTERFACING

Out of range values:

- **Pulse disturbances** (by inductors-like loads as motors, magnetic switches …) or random errors can occasionally give samples **very far** from the correct values.
- A good policy is to avoid the negative effect of samples that clearly are wrong due to random and macroscopic disturbs by replacing them with “**reasonable**” values.
- Often “reasonable” values means to take a value **equal to** the previous one (zero order maintaining) or preserving the **derivate** (first order maintaining)
- Other **finer approaches** exist to discard out of range values like as the difference compared to the previous value, or to the expected value, … and also in the choice of the value for replacing.
- This technique is often used **before** and **together** with filtering, averaging, …

ANALOG INTERFACING

Elaboration level (microprocessor/PC):

- **Managing regulation loops (control algorithms)**
- **Measurement display**
- **Printing, storing, time stamping**
- **Verifying threshold overtaking and alarms activation**
- **Data communication to other computers**