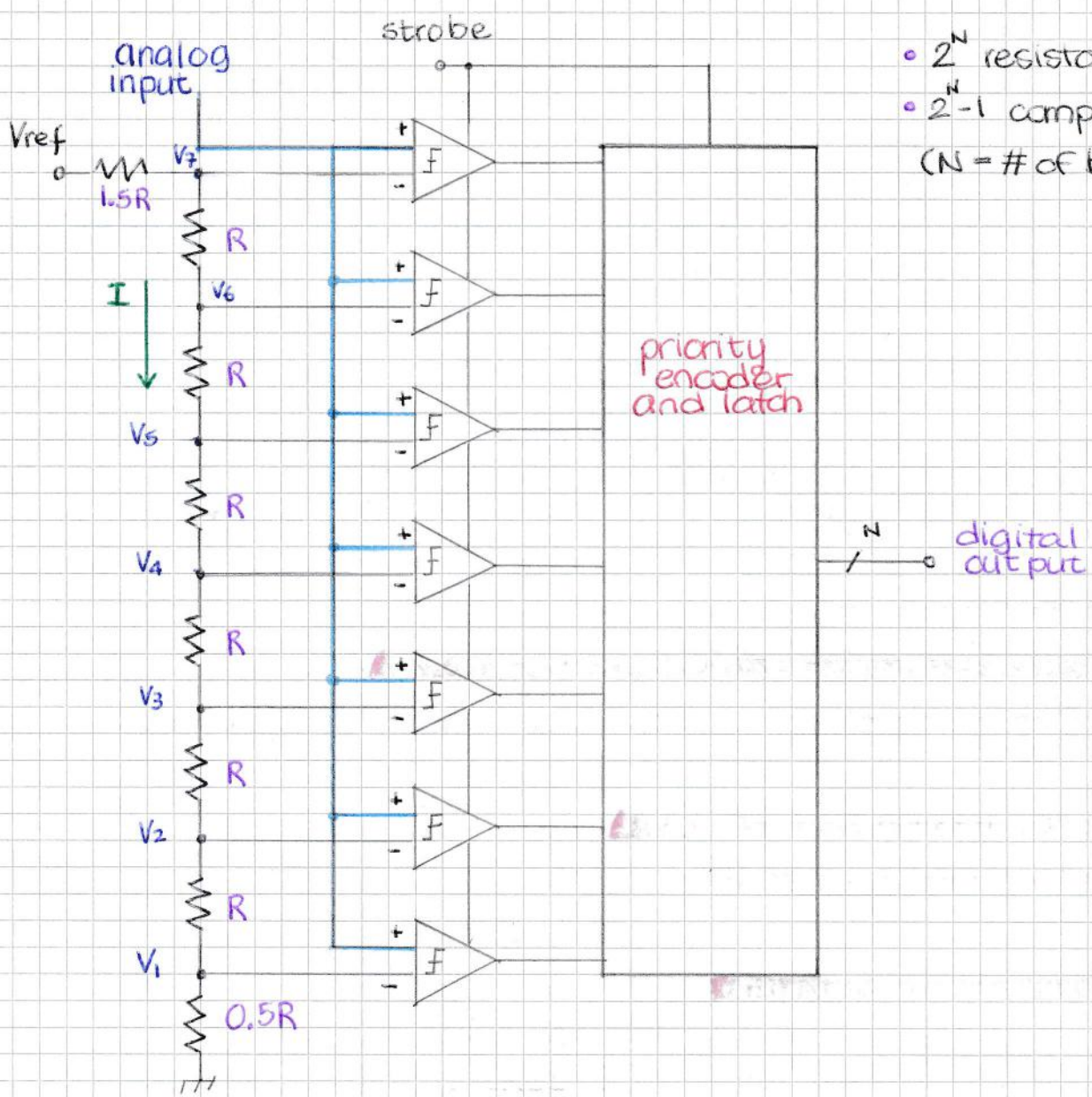
FLASH CONVERTER (Saletti) 02/11

- parallel ADC { bit computed at same time
  { fastest way to convert
  → large bandwidth
  → high power consumption
  → expansive
  → low resolution
  → occupies more space



- 2^N resistors
- $2^N - 1$ comparator
  (N = # of bits)

$Req = 1.5R + 6R + 0.5R = 8R$ (series of resistors)

$I = Vref / 8R$ => current flowing through the "ladder"

$V_1 = I \cdot \frac{1}{2}R = \frac{Vref}{16}$;

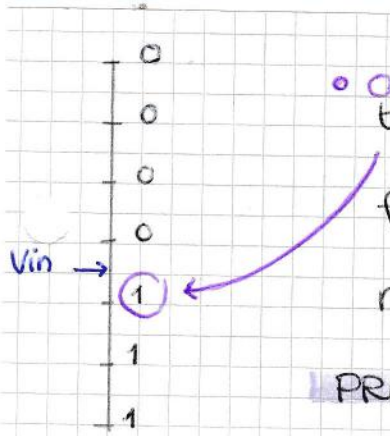
$V_2 = \left(R + \frac{1}{2}R\right) \cdot I = \frac{3}{16} Vref$

↑ equally spaced

these values get compared with the analog input signal

if $V_i < V_{input}$ => out = 1

=> the input signal is compared with every step of the ladder, in output (from the bottom) we have a series of "1" until a voltage is found that is higher than input signal so the comparator output is 0



- output of converter generated by selecting the top "1" in the set of threshold

find which is the last threshold that is smaller than Vin

↓

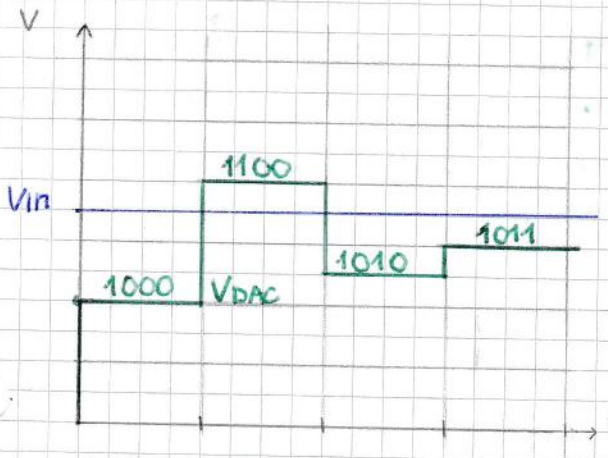
next threshold will be > Vin

PRIORITY ENCODER: selects the last "step" < Vin

SUCCESSIVE-APPROXIMATION ADC (SAR)

- most popular for data-acquisition applications
- resolution: 8 ÷ 18 bits
- sampling rate: up to several MHz (slower than flash ADC)
  - output generally provided with serial interface
- typically provided with sample & hold circuit to "freeze" input signal during conversion

successive approximation

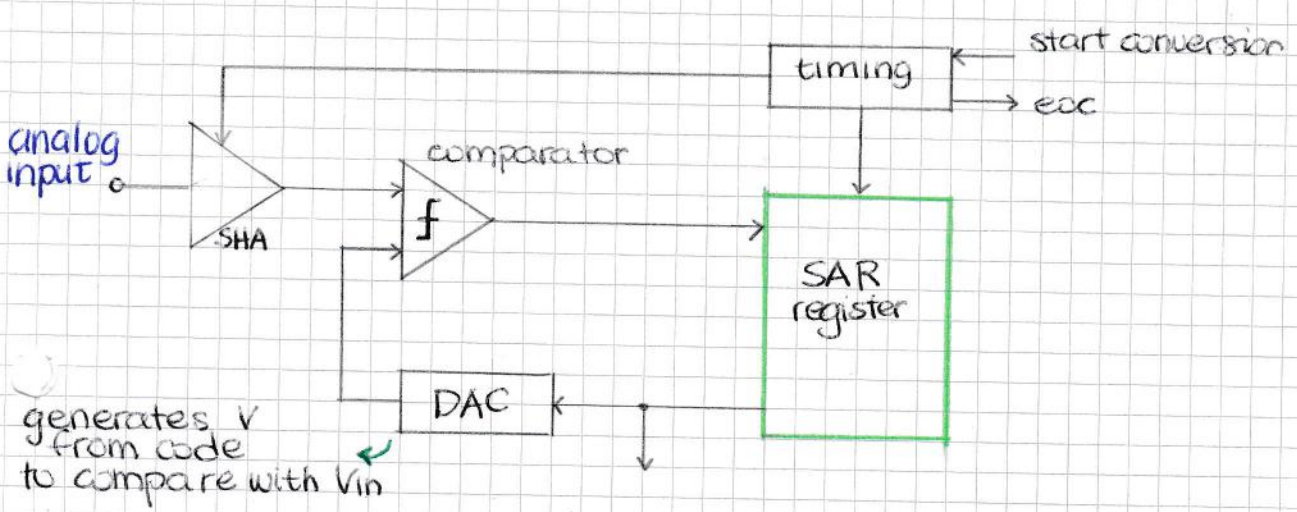


1) first bit = 1
2) if Vin > VDAC next step
   else first bit is changed to 0
3) successive bit = 1
   ...

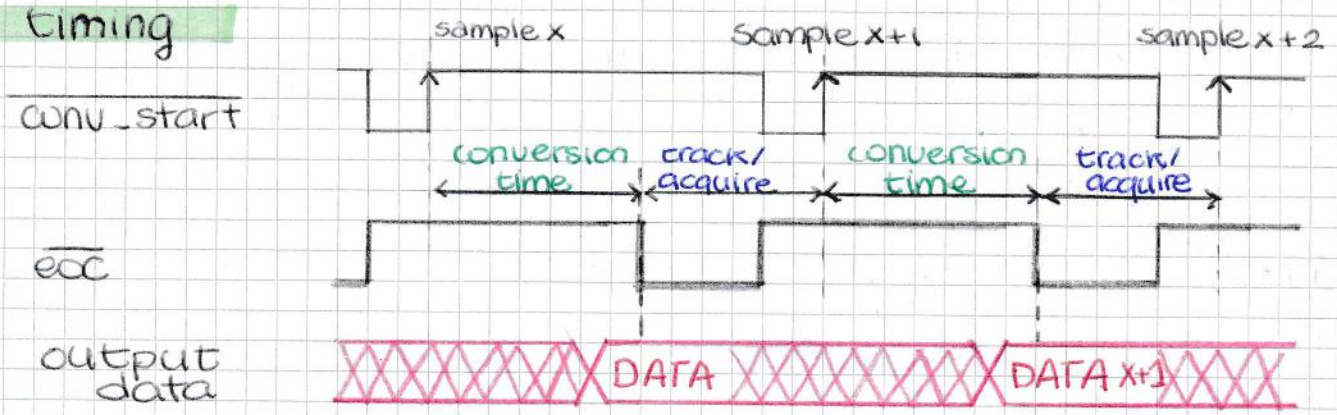
conversion is over after N steps (N = resolution)

since it takes time to convert value into VDAC and compare Vin must be constant (sample and hold)

architecture



timing

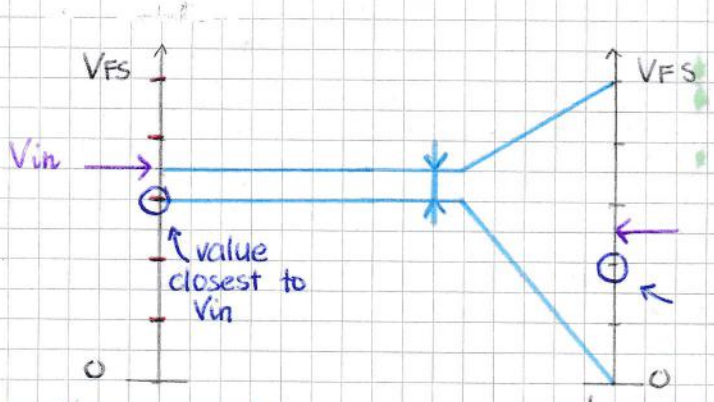


NOTE

- binary-tree-research alike conversion algorithm
- internal DAC set to midscale

TWO-STAGE SUBRANGING ADC (ex: 6bit converter)

two stages = 2 conversions { one coarse for 1st half / one finer for 2nd half

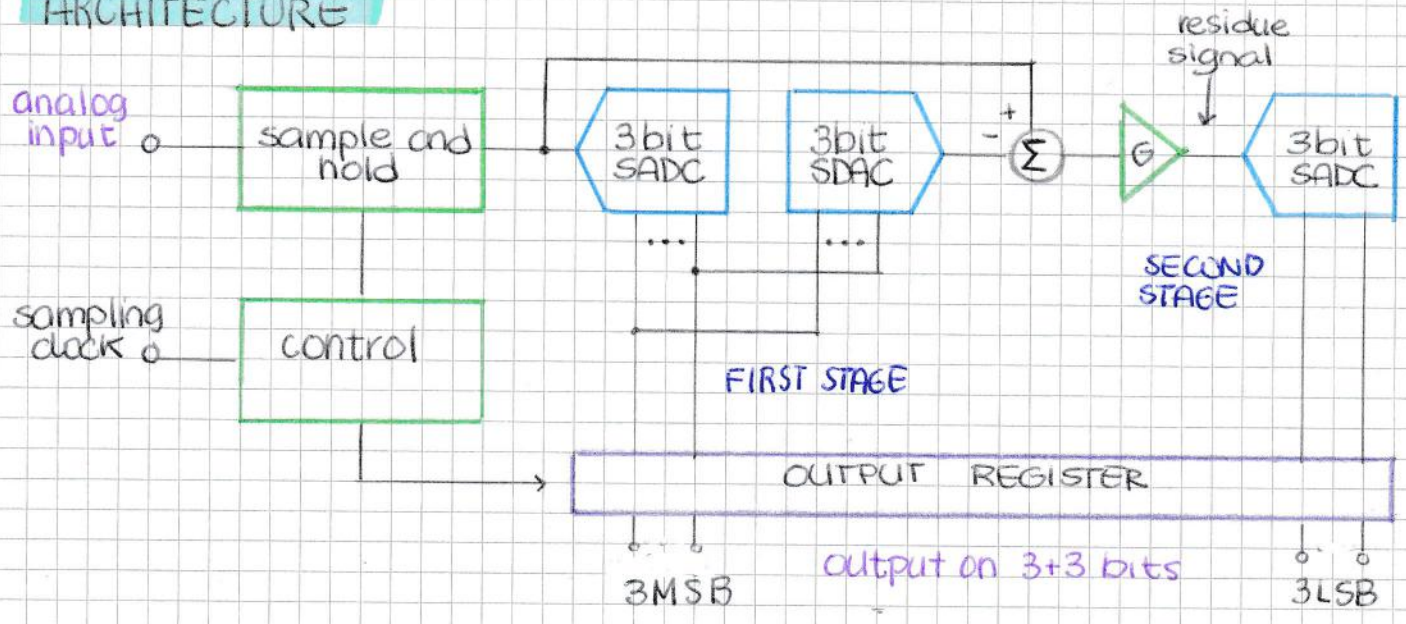


1) finds value closest to Vin (3
2) this value goes into a DAC
3) calculate Vin - VDAC
4) difference is amplified between 0 and Vref
5) convert the residual signal with 3 bits

1st stage
↳ coarse (first 3b)

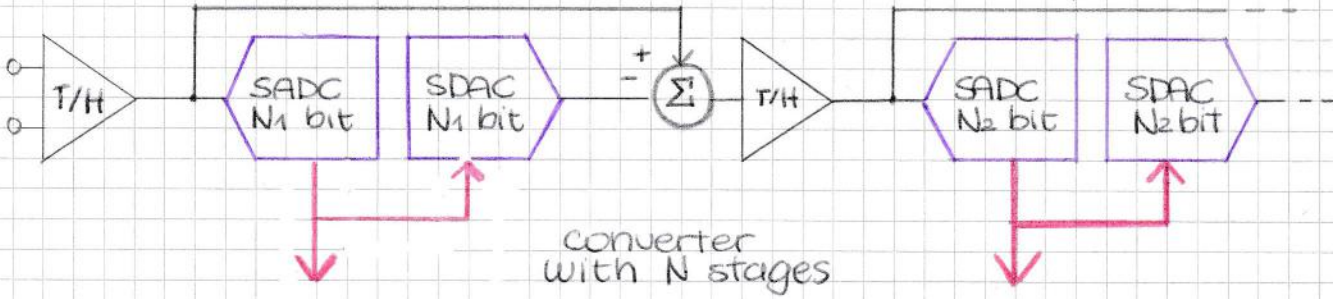
2nd stage
↳ fine (other 3b)

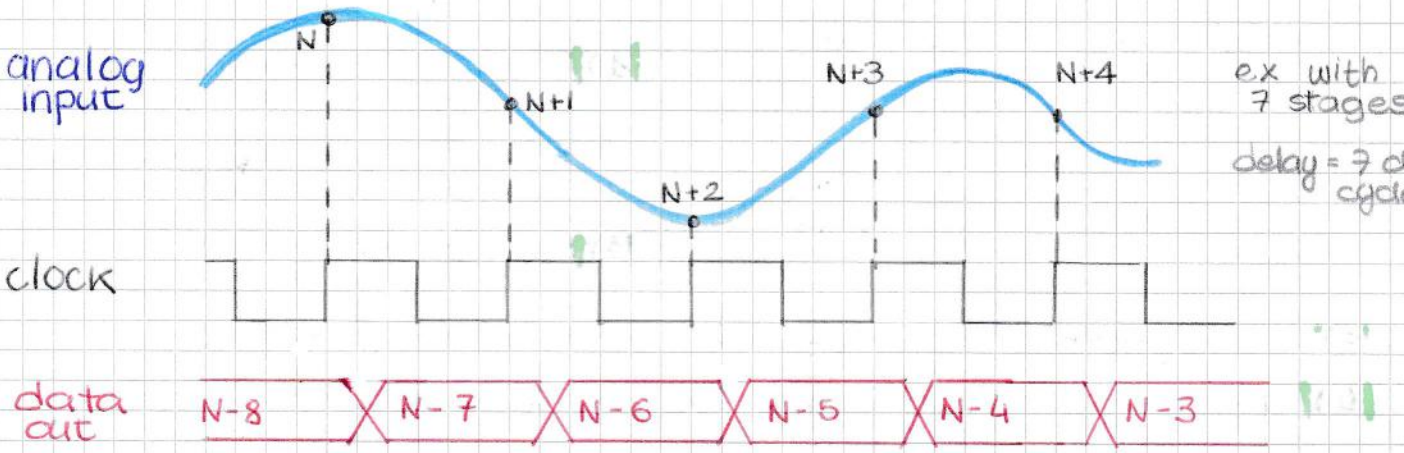
ARCHITECTURE



- sample and hold: "freezes" signal during conversion
- $2 \times 2^{N/2} - 1$ comparators needed (vs $2^N - 1$ in single stage)
  - less area and dissipated power (+)
  - lower conversion rate (-)

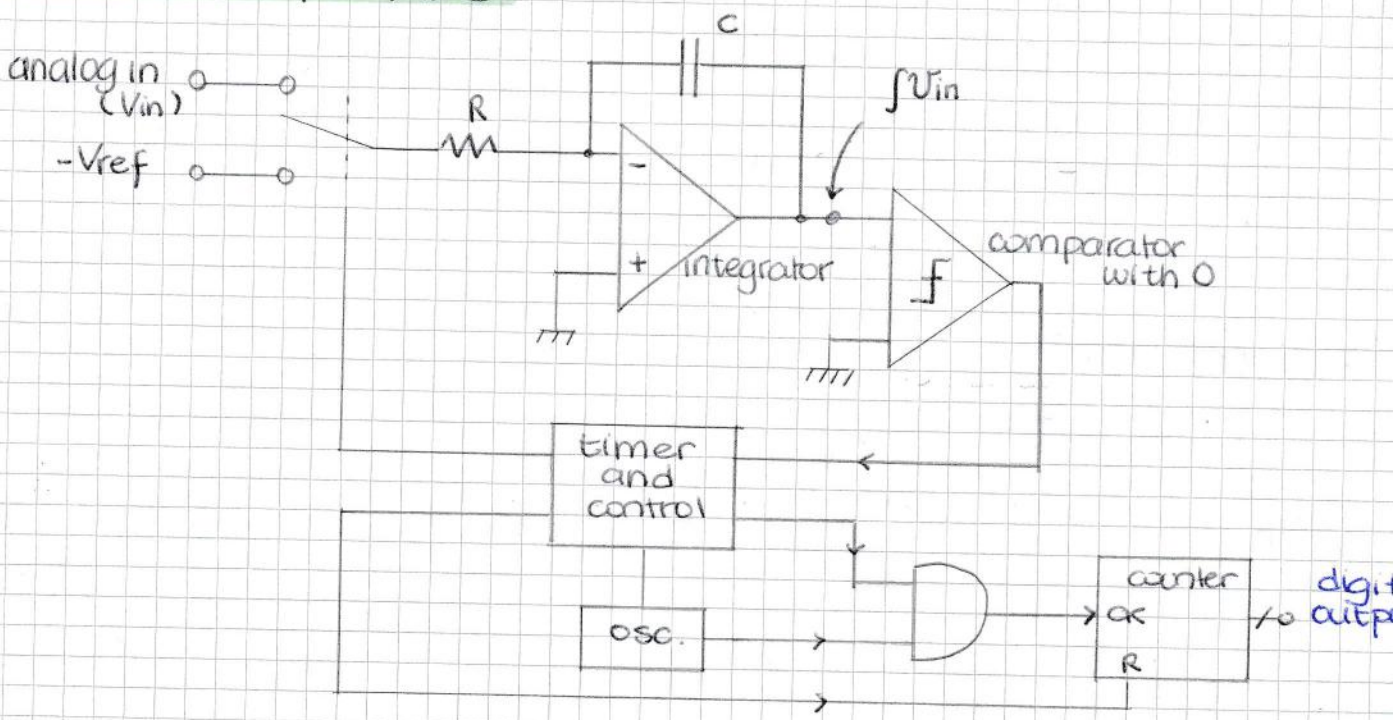
PIPELINE ADC

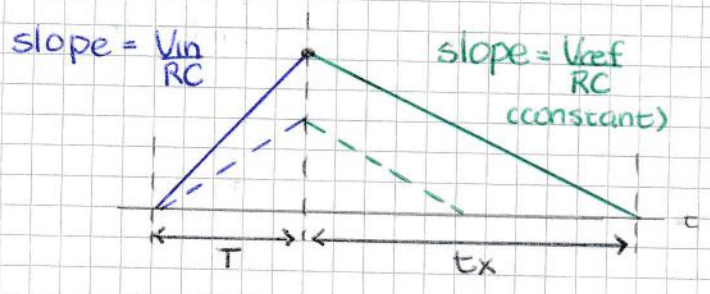




- at every time each stage of the converter operates on a different input value
- less time for conversion (+)
- to convert a single value it takes N clock cycle with N being the number of different stages (-)

DUAL SLOPE ADC





1) Vin is integrated for a known time T
↳ the 1st slope depends on the value of Vin

2) switch goes from Vin → -Vref

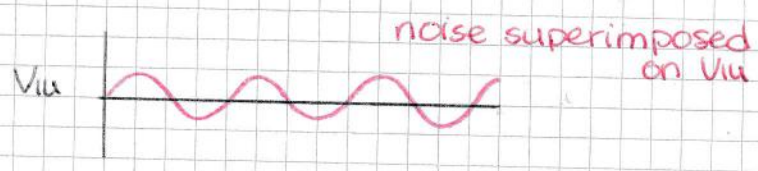
3) -Vref gets integrated → constant negative slope

4) after an unknown time tx the value reaches 0 and the computation ends

tx depends on the point reached by the first slope and it gets measured by the counter

$t_x = \frac{V_{in}}{V_{ref}} \cdot T$

noise effect



with sample and hold we may "freeze" at the wrong time (when noise is bigger compared to Vin)

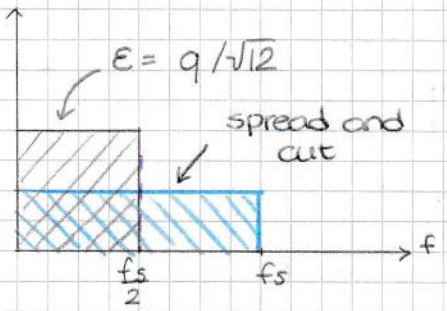
↓

we're integrating on the entire period of the NOISE itself so the NOISE gets canceled

=> integral calculation smooths down noise effect

PRINCIPLE of OVERSAMPLING

sampling input signal at a frequency much larger than the required one ($K \cdot f_s$)



→ without oversampling: quantization noise always present between 0 and $f_s/2$

→ with oversampling: quantization noise is spread between 0 and $K \cdot f_s/2$

→ by filtering out frequencies > $Kf_s/2$ the noise is largely reduced

GAIN = 3dB for each doubling of K

DECIMATION (DEC): we have way more samples than needed, this filter takes 1 sample every K

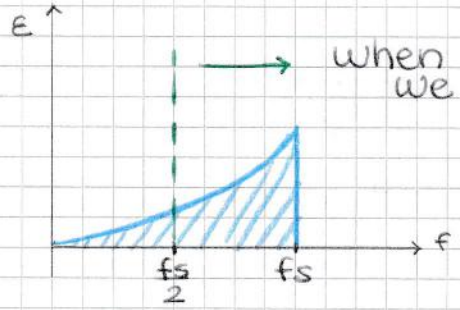
↳ must operate at a K times higher frequency

(+) noise reduction
(−) DEC filter must operate at high frequency

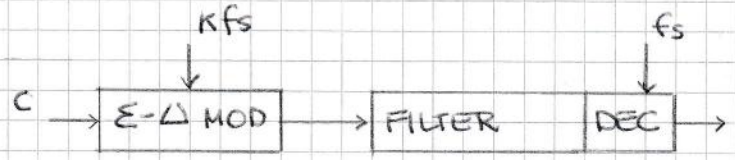
} need to find a balance when choosing K

Σ-Δ ADC BASIC PRINCIPLES

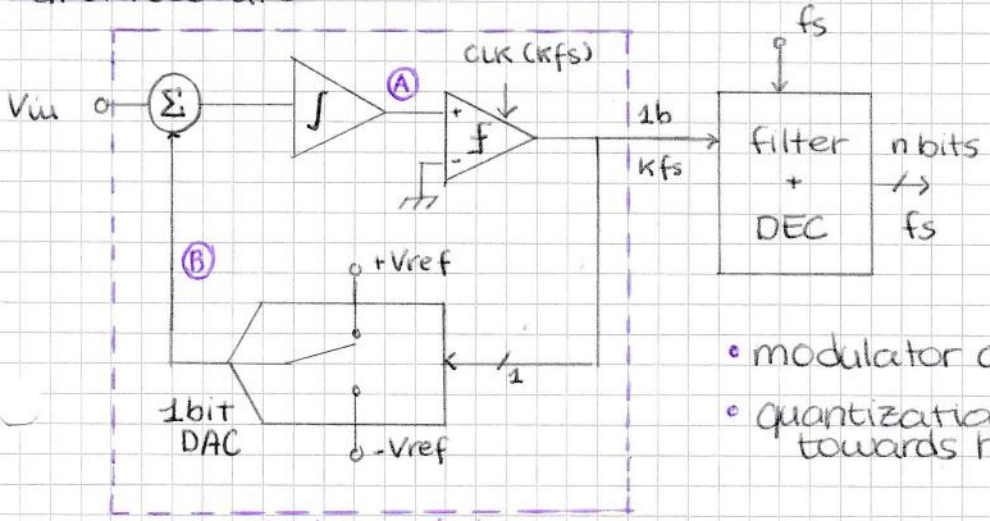
oversampling with Σ-Δ modulator to change the shape of noise by "pushing" it towards higher frequencies



→ when filtering high frequencies we remove more noise than before



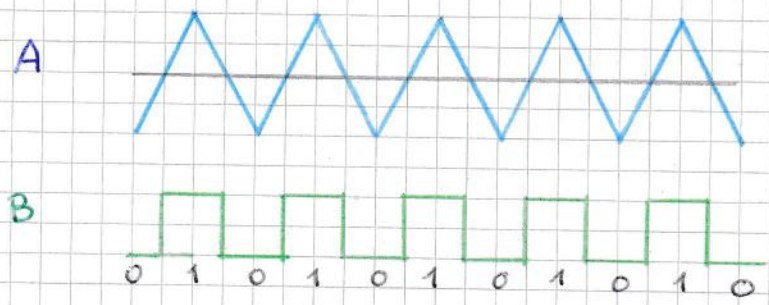
architecture



- modulator output = 1bit stream
- quantization noise shaped towards high frequency

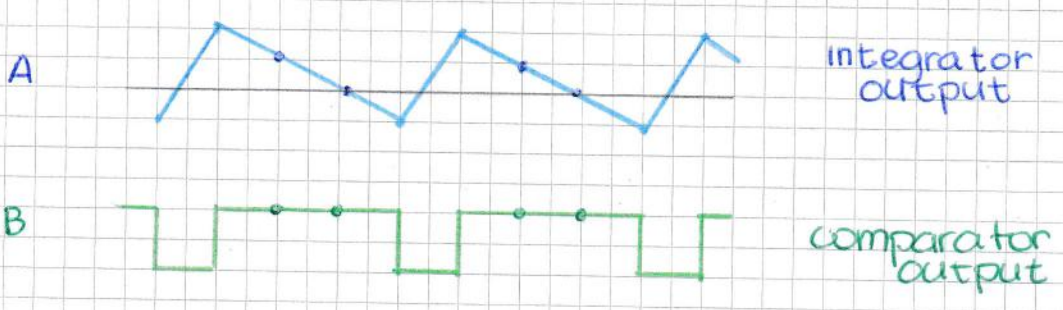
a) $V_{in} = 0$

- integrator receives $0 + V_{ref} = V_{ref}$ and then $0 - V_{ref} = -V_{ref}$
- integrator output = positive or negative ramp
- comparator gives 1 or 0 according to the ramp sign
- each time the sign of V_{ref} changes so the signal that gets integrated changes sign too



b) $V_{in} = V_{ref}/2$

- integrator receives $\begin{cases} V_{ref} + V_{ref}/2 = 3V_{ref}/2 \\ V_{ref}/2 - V_{ref} = -V_{ref}/2 \end{cases}$
- slope less steep than before, it takes 3 clocks to change from $1 \rightarrow 0$



- the number of "1" gives us information about V_{in}
- one result every interval decimates the oversampled sequence

a) $V_{in} = 0$

output seq: 0-1-0-1-0-1-0-1... => 8 "1"s over 16 samples

1 output every 16 samples => output = 1000

b) $V_{in} = V_{ref}/2$

output seq: 0-1-1-1-0-1-1-1... => 12 "1"s over 16 samples

1 output every 16 samples => output = 1100

(+) operations on 1 bit (simpler)

(+) fast, can oversample more, reducing noise even more

2nd ORDER Σ-Δ MODULATOR

- adds one integrator before the other one
- both get $V_{in} - V_{DAC}$ as input

(+) further improvement of SNR

Σ-Δ PROS AND CONS

(+) less noise

(+) 1 bit architecture

(+) most of computation is made on digital

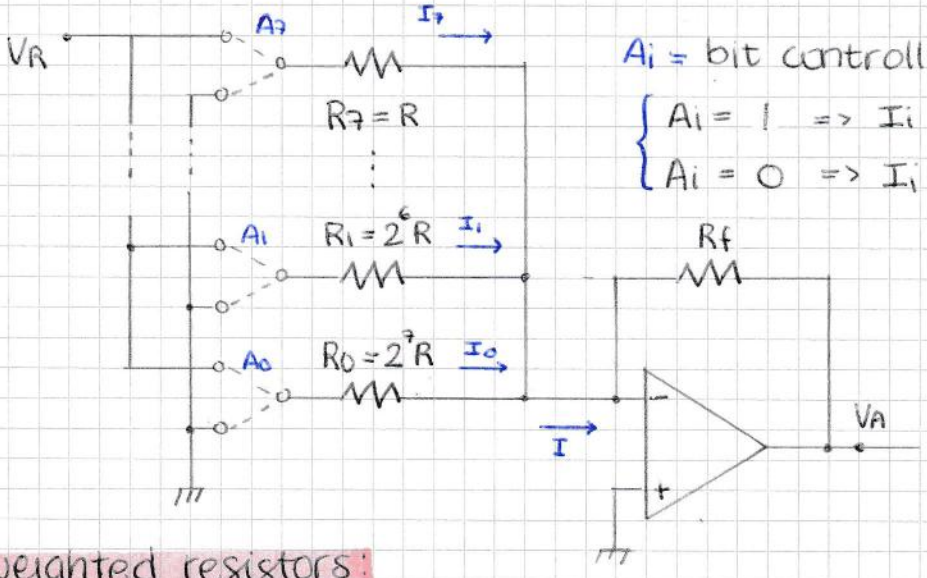
(-) oversampling requires high clock frequencies

(-) filtering takes several cycles causing latency between input and output

DIGITAL to ANALOG CONVERTERS (DAC)

typically used in motor control, we need to generate a voltage (or current) from sequences of bits

BINARY WEIGHTED RESISTORS



A_i = bit controlling the switch

$\begin{cases} A_i = 1 \Rightarrow I_i = V_R / R_i \\ A_i = 0 \Rightarrow I_i = 0 \end{cases}$

weighted resistors:

if the bit is more significative the current of the corresponding branch is bigger

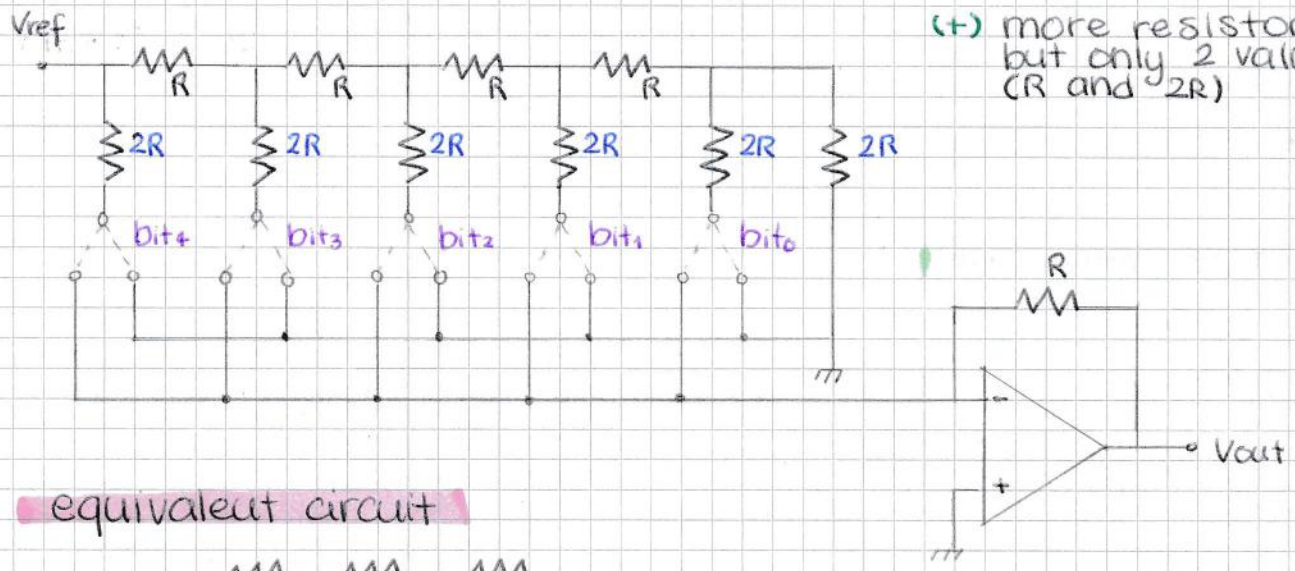
ex: $I_7 = V_r / R$; $I_0 = V_r / 2^7 R$

(-) different values of resistors

(+) simple architecture

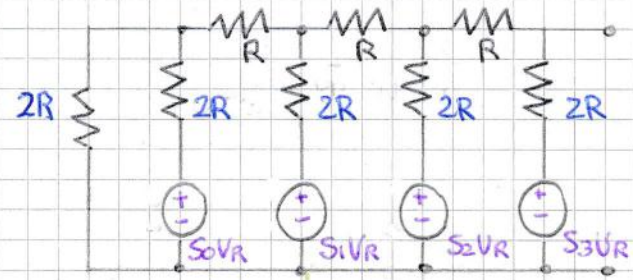
} good with low resolutions

R-2R LADDER RESISTORS



(+) more resistors but only 2 values (R and 2R)

equivalent circuit



- bits "control" voltage generators

V_{TH} (MSB → rightmost generator)

$V_{TH} = \frac{V_R}{2}$ (

V_{TH} (LSB → leftmost generator)

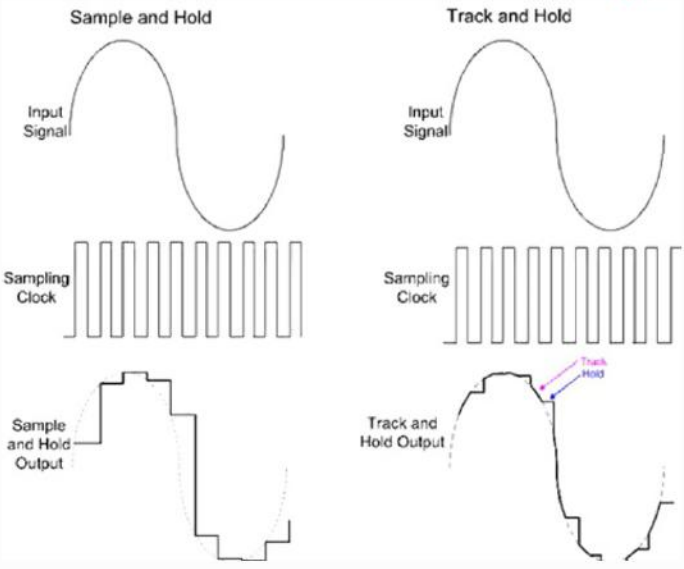
$V_{TH} = \frac{V_R}{16}$ → at each "step" V_R is divided by 2

the position of each generator determines its effect on the final voltage

↓

each bit has its own weight

DIFFERENZA TRA SAMPLE AND HOLD (S/H) E TRACK AND HOLD (T/H)



Entrambi i tipi di circuiti campionano il segnale di ingresso e mantengono costante la tensione campionata per la durata del processo di conversione. L'uscita del circuito T/H (a destra) traccia il segnale di ingresso finché non viene segnalato di effettuare il campionamento, quindi mantiene il valore di campionamento durante la conversione ADC. Il circuito S/H ha un'apertura di campionamento più breve e la sua uscita è una serie di livelli campionati (a sinistra). La differenza chiave tra T/H e S/H è la durata dell'intervallo di tracciamento: molto breve nel caso di S/H e molto più lunga nel caso di T/H. Entrambi i circuiti si basano su un commutatore veloce per isolare un condensatore di immagazzinaggio che è stato collegato all'ingresso del segnale