



Nyquist-rate A/D Converters

Pietro Andreani

Dept. of Electrical and Information Technology
Lund University, Sweden



- Introduction
- Timing accuracy
- Flash converters
- Sub-ranging and two-step converters
- Folding and interpolating converters
- Time-interleaved converters
- Successive-approximation converters
- Pipeline converters
- Other architectures

Introduction



The maximum frequency of operation of high-speed ADCs depends on the unity gain frequency f_{tech} of the used process, whose value determines the maximum obtainable f_T of an op-amp (assuming we are using op-amps, which is very often the case). f_T is lower than f_{tech} by a factor α , which is at least 2-4 (but ultimately depends on the desired accuracy)

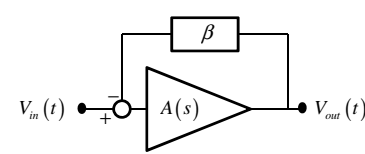
$$\alpha = \frac{f_{tech}}{f_T}$$

It is of course necessary to provide enough time for the analog signals to settle, thus requiring a suitable margin (γ) between the op-amp's f_T and the clock frequency.

Introduction – II



We continue by considering a feedback factor β around the op-amp (which is necessary to implement e.g. a gain stage):



$$A(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{A(s)}{1 + \beta A(s)} = \frac{\omega_T/s}{1 + \beta \omega_T/s} = \frac{\omega_T}{s + \beta \omega_T}$$

The (linear) time response to an input voltage step is:

$$V_{out}(t) = V_{in} (1 - e^{-t/\tau}); \quad \tau = \frac{1}{2\pi\beta f_T}$$

Introduction – III



An n -bit ADC needs an accuracy better than $2^{-(n+1)}$, which means that the settling time must be

$$e^{-t_{sett}/\tau} < 2^{-(n+1)} \rightarrow t_{sett} > \tau \cdot (n+1) \cdot \ln(2)$$

We may assume that the time allowed for settling is half a clock cycle; thus

$$\frac{1}{2f_{CK}} > t_{sett} \rightarrow f_{CK} < \frac{1}{2t_{sett}} = \frac{\pi\beta f_T}{(n+1)\ln(2)}$$

$$\gamma = \frac{f_T}{f_{CK}} > \frac{(n+1)\ln(2)}{\pi\beta}$$

Introduction – IV



For instance, if we have a 1.6GHz process, with $\alpha=2$ and $\beta=0.5$, we have

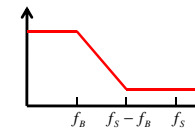
$$f_T = f_{tech}/\alpha = 800MHz \quad \tau = 1/(2\pi\beta f_T) = 0.398ns$$

For a 9-bit ADC requiring two clock periods per conversion we have

$$t_{sett} = 0.398ns \cdot (9+1) \cdot 0.693 \approx 2.7ns$$

$$f_{CK} = \frac{1}{2t_{sett}} \approx 164MHz \quad f_{sampling} = \frac{f_{CK}}{2} = 82MHz$$

With an anti-aliasing filter margin of one octave, the signal band is



$$f_s - f_B = 2f_B \rightarrow f_B = \frac{f_s}{3} = 27.5MHz$$

Timing accuracy

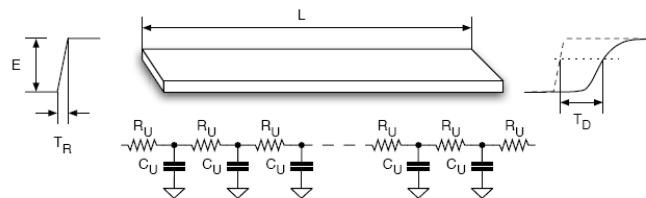


Recall the error due to sampling jitter:

$$\delta V_{in} = \delta T_{ji} \frac{dV_{in}}{dt}$$

The timing error must be less than $\frac{1}{2}LSB = V_{fs}/2^{n+1} \rightarrow$ a 12-bit ADC with 1V signal at 20MHz requires a jitter below 1ps!

Phase generation and distribution is also an issue, because of the distributed time constants in the metal wires



Timing accuracy - II



Assume a step with a finite slope,
$$e_i(t) = \begin{cases} \frac{E}{T_R}t & \text{for } 0 \leq t \leq T_R \\ E & \text{for } t > T_R \end{cases}$$

The step response is given by $e_{out}(t) = V_r(t) - V_r(t - T_D)$, with

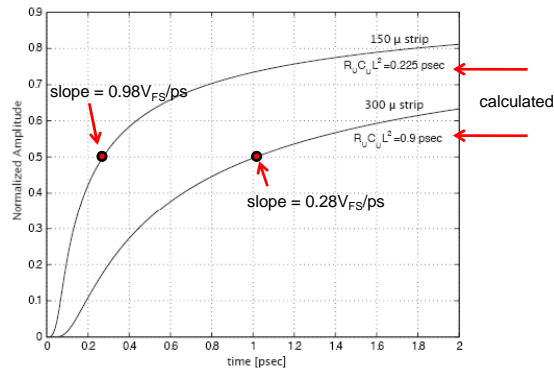
$$V_r(t) = \frac{E}{T_R} \left\{ (t + \tau/2) \operatorname{erf} \left(\sqrt{\frac{\tau}{4t}} \right) - \sqrt{\frac{\tau t}{\pi}} e^{-\frac{\tau}{4t}} \right\} \quad \tau = R_u C_u L^2$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$$

The delay is roughly estimated to be $T_D = \tau/4$, and it is easy to see that it does not depend on the width of the interconnection (if larger, C_u increases and R_u decreases by the same amount). Typically, $R_u C_u$ is in the order of a few tens of $as/\mu m^2$, leading to delays of 1ps over a few hundreds of microns of interconnection.

Example

$$R_u = 40 \text{ m}\Omega/\square; \quad C_u = 0.25 \text{ fF}/\mu\text{m}^2; \quad T_R = 50 \text{ fs}$$



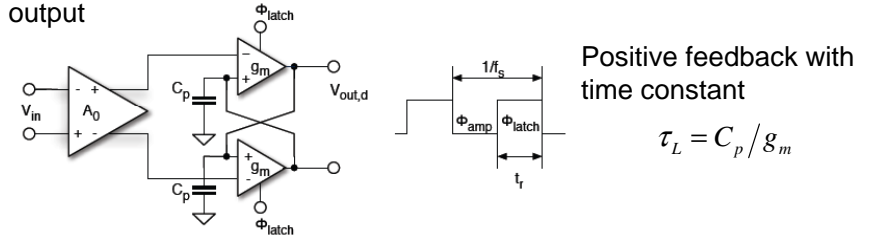
A noise of 40 mV_{FS} in the crossing threshold (0.5 normalized amplitude) results in a jitter of 40.8fs and 142.8fs, respectively; e.g.

$$\delta_{jit} = \frac{\Delta V}{slope} = \frac{40 \cdot 10^{-3}}{0.28} = 142.8 \text{ fs}$$

Metastability error

If the input V_{in} is not large enough, the comparator output may be undefined at the end of the latch phase \rightarrow error in the output code

Input signal \rightarrow pre-amplified during Sample and positive-feedback regeneration during Latch \rightarrow even a small input results in a "digital" output



Metastability probability error is $P_E \approx \frac{V_o}{V_{out,d}} = \frac{V_o}{A_o V_{in}} e^{-\frac{t_r}{\tau_L}}$, where:

V_o is the voltage swing for valid logic levels, and t_r is the period of the latch phase, typically equal to $1/2 f_s$

Metastability error – II

At reasonably low frequencies, the error probability is very low. The comparator must ensure an almost certain output for $V_{in} > 1/2 \text{ LSB} \rightarrow P_E$ must be lower than a given maximum for $V_{in} = V_{FS}/2^{n+1}$, leading to

$$f_s \ln \left(\frac{V_o \cdot 2^{n+1}}{P_{E,max} A_o V_{FS}} \right) \leq \frac{1}{2\tau_L}$$

For example, if

$$V_o \approx V_{FS}, \quad n = 8, \quad \tau_L = 2 \cdot 10^{-10}, \quad A_o = 10^3, \quad P_{E,max} = 10^{-4}$$

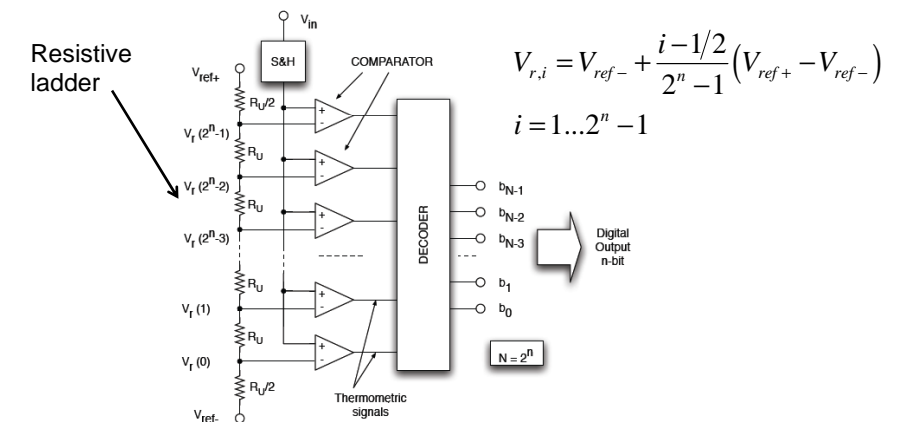
the maximum sampling frequency becomes

$$f_s = \frac{1}{17\tau_L} = 293 \text{ MHz}$$

Flash ADC

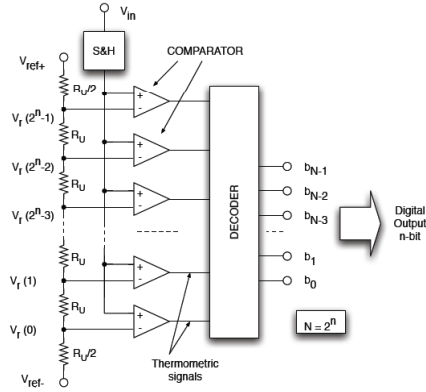
Fastest conversion: compares the input with $2^n - 1$ adjacent quantization levels; output is thermometric \rightarrow needs a decoder

Very high speed, low-to-medium resolution



Resistor values

Each comparator present a time varying load to the ladder → the unit resistance value must be low enough to allow the ladder to pull back all voltages values with an error below ½ LSB before the latch phase → simulation determine the optimal value for minimum power consumption

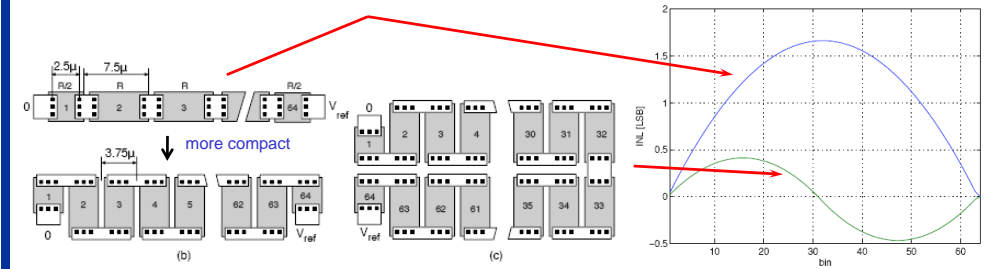


Distortion and INL

A linear resistivity gradient in the resistance ladder causes INL; the reduced cell pitch in b) is better than a), and even better is the folded layout in c)

Matching in modern technologies is 0.1-0.05% → 10-11 bits without trimming

Temperature drift across the ladder can cause INL as well (temperature coefficients as high as 10.000ppm/°C are possible)



Offset in comparators

The different offset at the input of two contiguous comparators changes the respective quantization interval from Δ to Δ_i :

$$\Delta \rightarrow \Delta_i = V_{thr,i} - V_{thr,i-1} = \Delta - (V_{os,i} - V_{os,i-1})$$

For example, an 8-bit flash with $1V_{FS}$ requires an $\sigma_{V_{os}} < 0.6mV$ for a 99.9% yield (i.e., $1/2^{8+1} > 3.3\sigma_{V_{os}}$)

The offset is mainly caused by the pre-amplifier in the comparator; if the transistors in the input differential pair are near-minimum length, the mismatch in threshold voltage and device length are much larger than those in width and conductivity; the input-referred total offset (usually, a few mV) becomes

$$V_{os} = \sqrt{\frac{A_{V_{thr}}^2}{WL} + \left(\frac{V_{od}}{2} \frac{\Delta L}{L}\right)^2} \quad \text{with} \quad \Delta V_{th} = \frac{A_{V_{th}}}{\sqrt{WL}}$$

$$\text{and} \quad \frac{I_{ds}}{g_m} = \frac{V_{gs} - V_{thr}}{2} = \frac{V_{od}}{2} \quad (\text{long-channel expression})$$

Offset in comparators – II

Explicit calculation for the offset introduced by ΔL : the current equation is

$$I_{ds} + \Delta I_{ds} = \frac{1}{2} \mu C_{ox} \frac{W}{L - \Delta L} V_{od}^2 \approx \frac{1}{2} \mu C_{ox} \frac{W}{L} \left(1 + \frac{\Delta L}{L}\right) V_{od}^2$$

$$\Delta I_{ds} = I_{ds} \frac{\Delta L}{L}$$

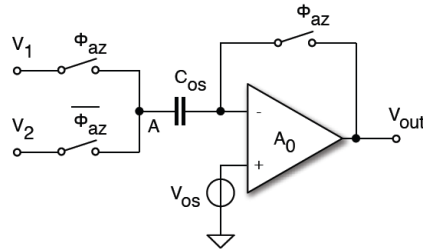
This is the error current at the output of the amplifier; since it is small, the relative ΔV_{in} at the input of the amplifier is calculated as

$$\Delta V_{in} = \frac{\Delta I_{ds}}{g_m} = \frac{I_{ds}}{g_m} \frac{\Delta L}{L} = \frac{V_{od}}{2} \frac{\Delta L}{L}$$

In bipolar implementations, the overall offset is often much lower than in CMOS (better intrinsic matching + exponential voltage-current relation)

Offset auto-zeroing

Offset is usually too large → offset cancellation is needed



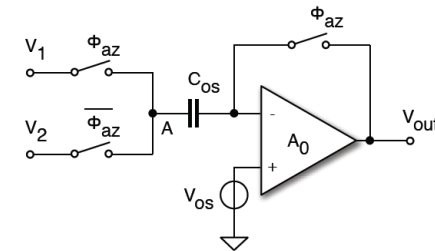
Two-phase approach → during Φ_{az1} , opamp in unity-gain configuration → C_{os} is charged to $V_{os} - V_1$; during the second phase, we have

$$V_{diff, in} = V_+ - V_- = V_{os} (\overline{\Phi_{az}}) - V_2 - (V_{os} (\Phi_{az}) - V_1)$$

If the offset does not change → perfect cancellation; also 1/f noise is removed (white noise is instead doubled, because uncorrelated)

Charge injection is an issue (switch in feedback path critical); however, offsets of fraction of mV are possible (with fully differential architectures)

Offset auto-zeroing – II

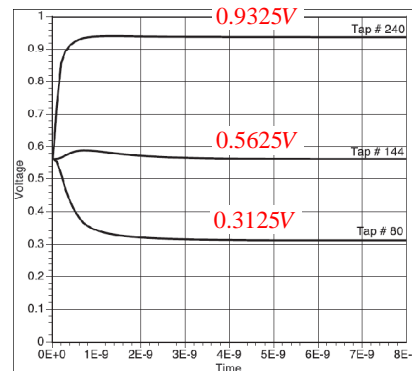
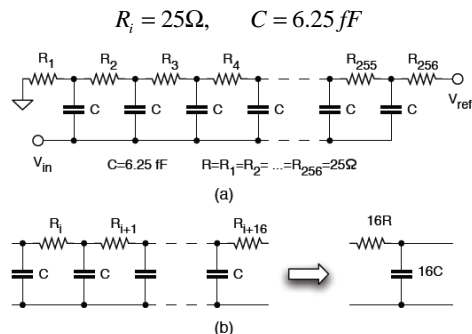


One terminal is the input of the flash, the other is the tap from the ladder → the difference between input and reference is not calculated by a differential pair, but by the previous voltage subtraction at node A!

The par. cap. between A and GND is charged to V_1 and then to V_2 → the ladder must charge/discharge quickly enough these capacitances (slowest taps at the middle of the ladder)

Example – ladder settling

Simplified example ($n=8$) → tap #80 takes ~5.6ns to settle to less than $\frac{1}{2}$ LSB (of its nominal level of 0.3125V) after converting an input of 0.5625V → if this is too long, and the next input is close to 0.3125V, there could be a code error – notice also that tap #144 is not constant even if 0.5625V is its nominal value – this is because the ladder must (dis)charge all other nodes as well



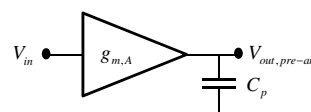
Practical limits

The speed and resolution of flash converters depend on a number of practical limits:

- 1) Small unit resistors in the resistive divider → improve resolution and speed, but require higher power and a voltage reference with low impedance from DC to f_s → very difficult to realize
- 2) Exponential increase in complexity and power consumption with # bits (especially power cannot grow beyond budget)
- 3) Effectiveness of comparators, in terms of metastability probability error → at high frequencies, the gain of the pre-amp (i.e., ratio of output signal (at the end of pre-amp phase) and input signal) does not have enough time to reach its low-frequency value

Practical limits – II

At high frequencies, the “dynamic” gain of the pre-amp is determined by the current delivered by the pre-amp to C_p , yielding a voltage ramp across Φ_{amp} (without reaching its maximum value) \rightarrow the final value of the output voltage is



$$V_{out,pre-amp} = (V_{in} g_{m,A}) \frac{1}{C_p} \cdot \frac{T_{ck}}{2} = \frac{V_{in} g_{m,A}}{2 f_{ck} C_p} \rightarrow A_0 = \frac{g_{m,A}}{2 f_{ck} C_p}$$

More bits \rightarrow more gain needed \rightarrow more $g_{m,A} \rightarrow$ much more current (x4 for 1 more bit, or for doubling the sampling speed)

Example: 7-bit 500MHz flash that needs a pre-amp gain of 20; MOS overdrive $V_{od}=200\text{mV}$, $C_p=0.4\text{pF}$. Since $g_m = 2I_{ds}/V_{od}$, we obtain for the diff-pair

$$I_{diff-pair} = 2I_{ds} = g_m V_{od} = 2A_0 f_{ck} C_p \cdot V_{od} = 1.6\text{mA}$$

which means that 200mA are needed only in the pre-amp stages

Practical limits – III

Another important limitation is due to the capacitive load on the input S&H from the par. caps of all comparators, $2^n C_p$

The charge on this cap. after sampling the reference is $2^n C_p V_{ref} / 2$

A full-scale input voltage drains an equal amount of charge from the S&H; this charge must be provided in a fraction α of the sampling period; the peak current that the S&H must deliver is

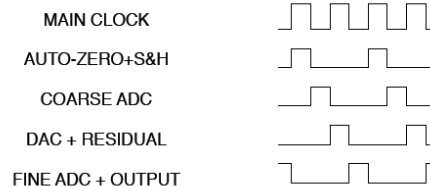
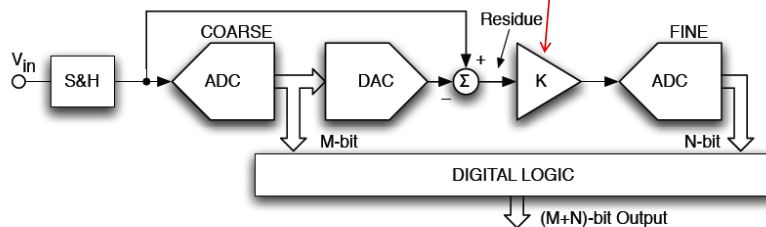
$$I_{S\&H,pk} > f_s 2^n C_p \Delta V_{in,max} / 2\alpha$$

which is easily larger than 10mA.

To summarize, it is impractical to design an 8-bit flash with $> 500\text{MS/s}$, or a 6-bit flash with $> 2\text{GS/s}$ (these numbers improve along with the CMOS technology, of course)

Sub-ranging / Two-step converter

$K=1 \rightarrow$ sub-ranging; $K>1$ (e.g. 2^M) \rightarrow two-step



possible clocking scheme (fast)

Advantages

comparators is much reduced: if 8-bit and $M=N=4$, we need $2 \cdot (2^4 - 1) = 30$ comparator instead of $2^8 - 1 = 255$ in the full flash!

The spared area and power are much more than what is needed in the DAC and residue generator – if $K=2^M$, the dynamic range of the amplified residue equals that of the input signal \rightarrow coarse and fine ADCs can share the same reference voltage

S&H loaded only by $2^M - 1$ comparators

Conversion rate is in principle reduced, as 2-3 clock periods are needed per conversion

However, since the speed of the S&H is the bottleneck in medium-resolution full-flash ADCs, the clock of the sub-ranging can actually end up being at higher frequencies, as the reduced capacitive loading of the S&H enables a faster S&H

Accuracy requirements

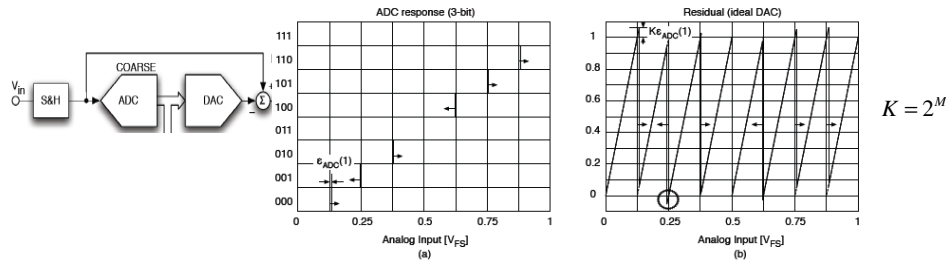
For medium resolutions, the quantization step is 1mV or more → the capacitance required to keep kT/C smaller than $\frac{1}{2}$ LSB is not very large

For instance, if $C=0.5pF$, $\sqrt{kT/C} = 90\mu V$ → up to 10b, the input capacitance is not a problem in the S&H design

The residue is given by

$$V_{res}(V_{in}) = K(V_{in} - V_{DAC}(i)) \text{ for } V_{coarse}(i-1) < V_{in} < V_{coarse}(i)$$

Ideally, the residue is a saw-tooth with amplitude between 0 and $V_{FS} \cdot K/2^M$
Below is instead the residue with a real coarse ADC and an ideal DAC



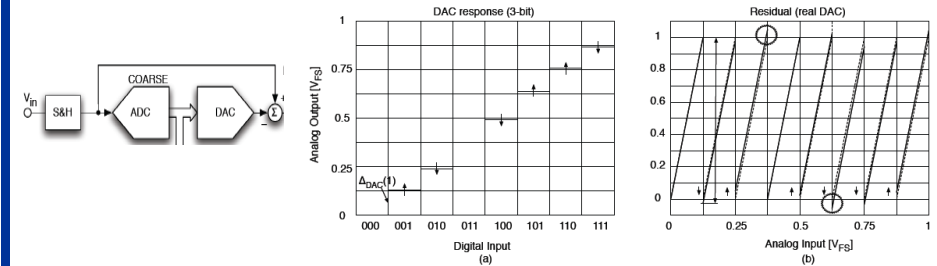
$$K = 2^M$$

Accuracy requirements – II

A residue outside the range of the LSB ADC results in all zeros (ones) until the input re-enters the boundaries

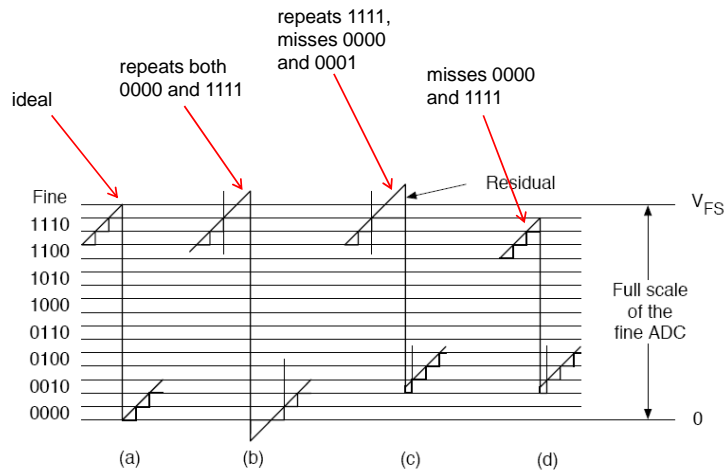
The DAC generates the subtractive term → a DAC error alters the residue as in the figure below (where the ADC is ideal)

Important: shift caused by DAC lasts an entire “tooth” → accuracy demands on the DAC are more stringent than on the ADC (whose errors can be corrected, as they are localized around the break points - either by inserting extra thresholds outside the $0-V_{FS}$ region, or with the same technique that will be discussed later within pipeline ADCs)



Residue conversion

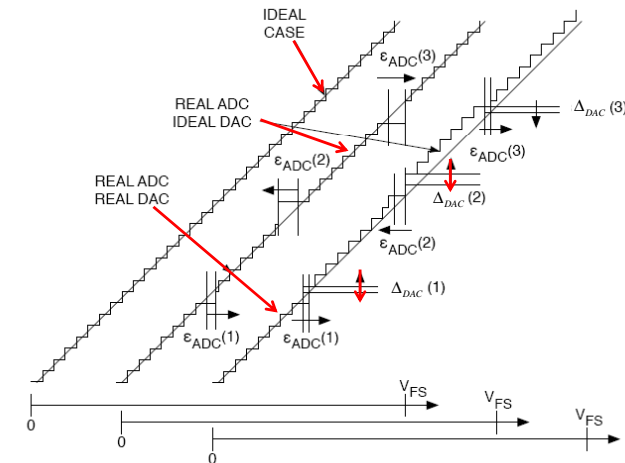
Possible situations for the fine conversion, with 4-bit LSB – depending on the errors affecting the residue



Overall picture

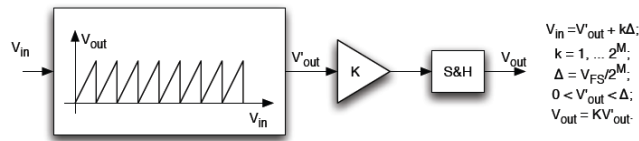
ADC errors (middle curve) → only around MSB transitions; after 1-2 LSB from MSB transitions, response is again on the interpolating line

DAC errors → affect whole LSB range



Two-step as a non-linear conversion

The generation of the residue is equivalent to the action of a non-linear block



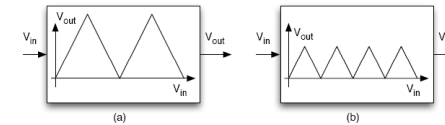
Non-linear operations alter the signal spectrum by generating extra tones \rightarrow in data-sampled systems, a narrow-band input can spread over the Nyquist range \rightarrow amplifier and LSB flash ADC must work well above Nyquist in order to avoid degradation in the LSB conversion

Spectrum of the residue is only weakly correlated with the input (provided the amplitude is a few MSBs at least) \rightarrow distortion in the LSB blocks look more like white noise than tones \rightarrow linearity of the residue generator is not critical \rightarrow SNDR and SFDR set by first ADC and DAC

Folding

Two-step \rightarrow dynamic range divided into MSBs, with linear input-output relations inside each MSB

An equivalent non-linear transformation is the folding of a straight line shown below \rightarrow folding once around $V_{FS}/2$: 1 bit; twice around $V_{FS}/4$: 2 bits; three times around $V_{FS}/8$: 3 bits; and so on



The number of intervals required to quantize the folded signal diminishes accordingly \rightarrow after an M-bit folding, only $2^{N-M}-1$ comparators are needed for the N-bit conversion

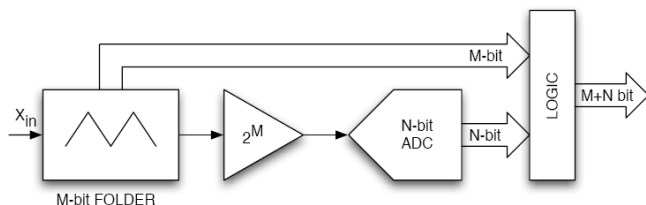
Obviously, it is necessary to know from which folded segment the input is coming, to determine the MSBs

Folding – II

The M-bit folder produces two signals: the analog folded output, and the M-bit code identifying the segment used in the folded response

The gain stage possibly boosts the dynamic range of the analog folded output to V_{FS}

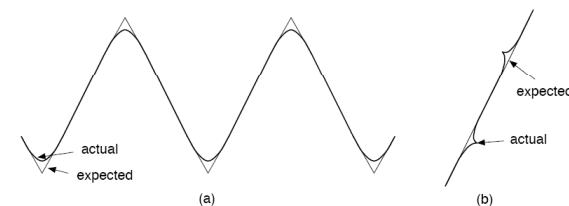
The N-bit ADC determines the LSBs, and finally the logic block combines MSBs and LSBs to deliver N+M bits



Real folding

Any circuit implementing folding is not able to achieve infinitely sharp transitions – corners are always more or less rounded

The error is estimated by unfolding the folded response – monotonicity is sure, but INL may be high



In general, working on different segments results in different delays

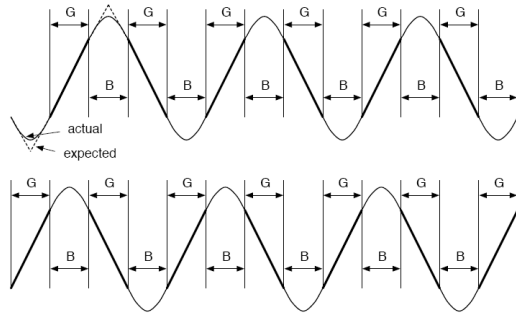
Folding is normally used for high conversion rates and medium-high resolutions \rightarrow finite bandwidth and slew-rate in folding block are crucial

Double folding

Double folding avoids the use of the non-linear regions present in simple folding → higher linearity, but also higher complexity

Two transfer characteristics, out-of-phase by ¼ of the folding period → one is always in the linear region

The logic must decide which of the two characteristics should be used



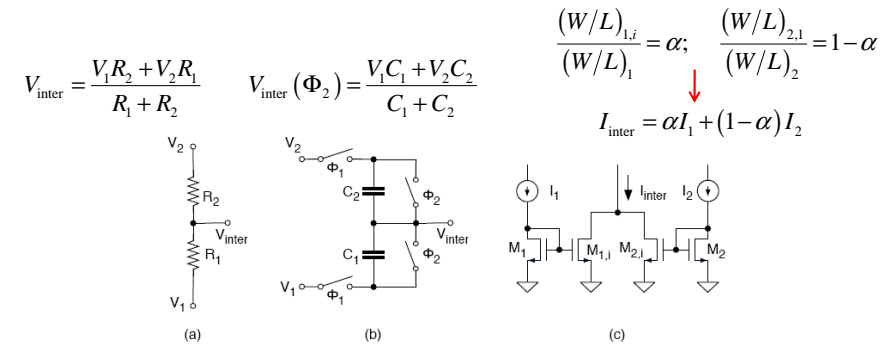
Interpolation

Interpolation → value intermediate between two other values

With voltages → implemented with resistors, or capacitors → a) and b) below

With currents → implemented with current mirrors → c)

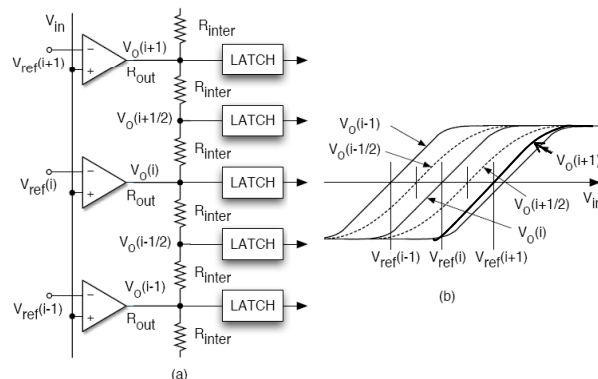
Accuracies of 0.1% can be expected with good layout



Interpolation in flash converters

Comparators substituted with pre-amp; reduces # pre-amps and ref. voltages (but not # latches) → cap. load on the S&H diminishes → less power consumption, higher speed; fewer ref. voltages → less "charge-pumping" effect → less stringent settling limit

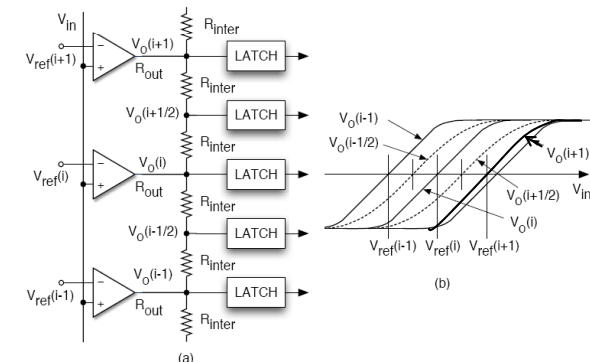
Often used with 4 or 8 resistors (not just 2 as here)



Interpolation in flash converters - II

To preserve linearity, pre-amp output should saturate for an input higher than the closest upper threshold, and lower than the closest lower threshold.

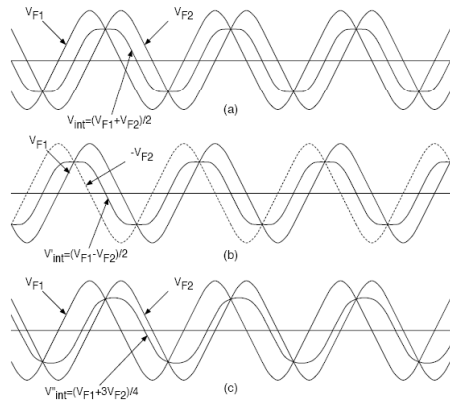
The overlapped non-saturated regions, interpolated by the resistors, determine zero-crossings mid-way between the pre-amps zero crossings. Far from the zero crossings, the slope of the interpolated curve diminishes, but then the differential signal is already large enough for the latch.



Interpolation with folding



Multiple interpolators take the place of the fine flash converter → (a) shows the interpolation between two folded responses (V_{F1} and V_{F2}), shifted by half segment; the shape of the interpolated segment is more rounded than the generating signals; but, again, it is important that it is linear only close to the zero crossing



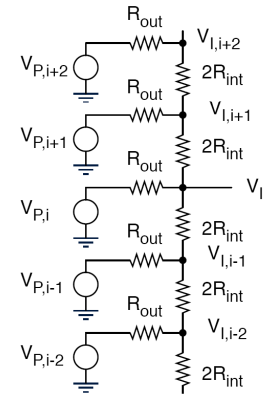
(b) → interpolating V_{F1} and $-V_{F2}$ yields a second set of zero crossings, detected by additional comparators

(c) → if $R_1 = 3R_2$, the zero crossing is at $1/4$ distance from the zero crossing of V_{F2} → multiple interpolations can yield a sufficient number of crossings to obtain the LSB conversion → avoids the use of an explicit flash ADC, possibly increasing the conversion rate

Interpolation and linearity



Interpolation resistances load the pre-amps → if output resistance is not much lower, voltage drop alters the generated voltages → however, this is in fact an advantage! For the first, the voltage error depends on the generated voltage, and is zero for zero output voltage → influential at zero crossing, which is where it really matters!



Besides this, interpolating resistors + pre-amps with finite output resistance average the offsets of the pre-amps, improving the overall linearity
The current flowing into the i -th pre-amp is

$$I_{P,i} = \frac{V_{I,i+1} + V_{I,i-1} - 2V_{I,i}}{2R_{int}}$$

→ if the i -th interpolated voltage is the average of its neighbors, its current is zero → ideal output voltages are not affected by the interpolating network → we need to consider only the effect of the errors

Interpolation and linearity – II



We assume an error on the output voltage of the i -th op-amp: $V_{P,i} \rightarrow V_{P,i} + \epsilon_i$
The error at the output i becomes

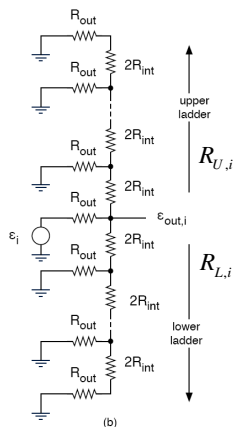
$$\epsilon_{out,i} = \epsilon_i \frac{1}{1 + R_{out}/R_{T,i}} = \epsilon_i T_{i,i} < \epsilon_i; \quad R_{T,i} = R_{U,i} \parallel R_{L,i}$$

where $T_{i,i}$ is a damping factor introduced by the finite output resistance of the pre-amp

Using the circuit on the left, it is possible to estimate the error induced by ϵ_i at a node j ; $T_{i,j}$ diminishes with $|i - j|$ increasing, and is usually negligible for $|i - j| > 4$. In general, we must superpose the effects of all errors:

$$\epsilon_{tot,i} = \sum_{j=1}^{2^n-1} \epsilon_j T_{i,j}$$

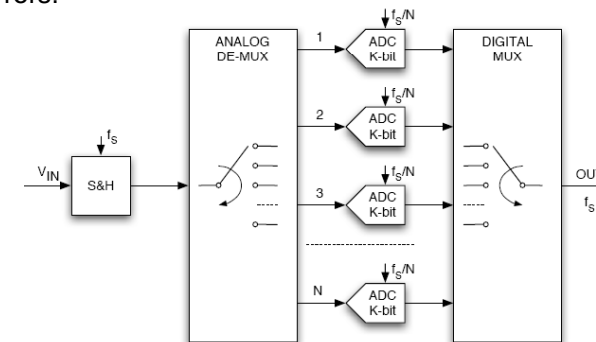
The above averaging effect benefits short-distance errors → reduces the DNL



Time-interleaved converters



Increase the conversion rate by having N converters working in parallel → the equivalent conversion rate is N times that of the single converter; the S&H must work at full rate; or, there can be $\#N$ S&H working at the reduced rate, but with a very high precision in the distribution of the phases. Important remark: gain and offset errors on each ADC, which are usually of minor weight, are very important here, because they become dynamic errors!



Accuracy requirements



Clock misalignment across the ADCs → looks like clock jitter, but with $N \cdot T_S$ periodicity. If the clock misalignment between the K -th and 1st channel is δ_k , the error introduced is

$$\varepsilon_{ck,K}(nT) = \delta_K \left. \frac{dV_{in}}{dt} \right|_{nT}; \quad n = iN + K$$

which, for the sinusoidal input $V_{in} = A_m \sin(\omega_m t)$, becomes

$$\varepsilon_{ck,K}(nT) = \delta_K A_m \omega_m \cos(\omega_m nT)$$

The power of the error due to clock misalignment becomes, as a function of the input power and N :

$$P_{\varepsilon_{ck}} = P_{in} \delta_k^2 \omega_m^2 / N$$

Furthermore, because of the down-sampling of the input with a clock error at frequency f_s/N , the clock misalignment gives rise to images located at

$$\frac{kf_s}{N} \pm f_{in}$$

Accuracy requirements – offset



Offset → identical offsets in all ADCs have no impact; offset mismatch causes tones → offset on a signal channel causes a pulse of amplitude equal to the offset → tones at f_s/N and its multiples

Interleaved converter is for large bandwidths → signal occupies a large fraction of the Nyquist interval → even tones at relatively high frequencies are likely to fall into the signal band

Worst case → alternate sequence of positive and negative offsets V_{os} → square wave of amplitude V_{os} → 1st harmonic with amplitude $4V_{os}/\pi$; SFDR with full-scale input sine wave becomes

$$SFDR = 20 \log \frac{\pi V_{FS}}{8V_{os}}$$

If the SFDR requirement for maximum input equals the SNR, the offset mismatch should obey the relation

$$\frac{1}{2} \left(\frac{4}{\pi} V_{os} \right)^2 \leq \frac{\Delta^2}{12} \quad \rightarrow \quad V_{os} \leq \frac{\pi}{4} \sqrt{\frac{1}{6}} \cdot \Delta = 0.32\Delta = 0.32LSB$$

Accuracy requirements – gain error



Identical gain errors in all ADCs have no impact; gain mismatch between the channels causes tones → worst case if gains alternate between $(1+\varepsilon_G)$ and $(1-\varepsilon_G)$, which gives rise to an error equal to the multiplication of the input signal with a square wave of amplitude $2\varepsilon_G$ at $f_s/2$ (or its sub-multiples)

Largest spur tone occurs at $f_s/2 \pm f_{in}$ (or $f_s/4 \pm f_{in}, \dots$), and has amplitude

$$A_{spur} = \frac{4}{\pi} \varepsilon_G A_m$$

resulting in

$$SFDR = 20 \log \frac{\pi}{4\varepsilon_G}$$

Thus, even with ε_G as low as 0.1%, the SFDR is not higher than 58dB

Bottom-line: time-interleaved ADCs always need trimming/calibration if high resolution is desired

Successive-approximation converter

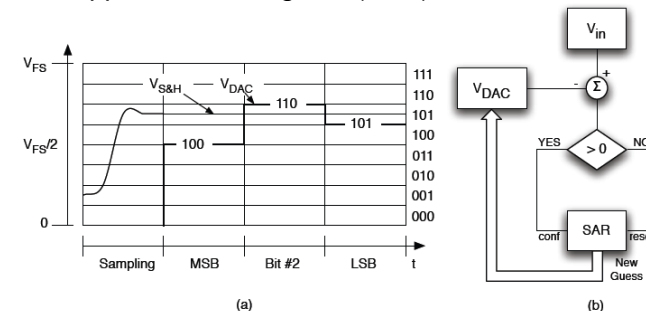


1 bit converted per clock cycle (+ 1 or 2 cycles for signal sampling + settling)

Binary search → reduced complexity and power, but lower conversion rate

The MSB distinguishes between input signals that are above or below $V_{FS}/2$; depending on this result, the threshold for determining the 2nd bit is either $V_{FS}/4$ or $3V_{FS}/4$; and so on (below: 3-b conversion)

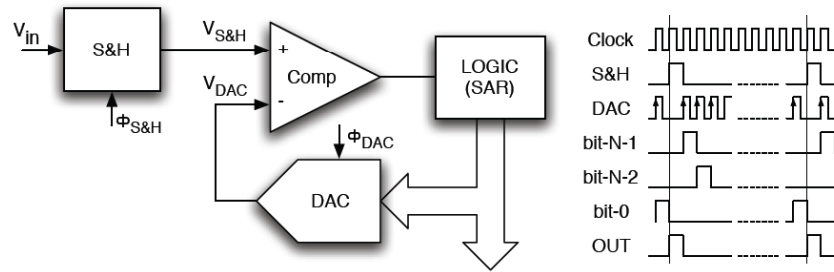
The voltages used in the comparisons are generated by a DAC driven by a successive-approximation register (SAR)



SA – timing diagram



After signal S&H, the SAR sets the MSB to 1; if the comparator confirms, the 1 is retained, otherwise is set to zero; then the 2nd bit is processed, and so on until all bits have been generated; then a new signal sample is taken

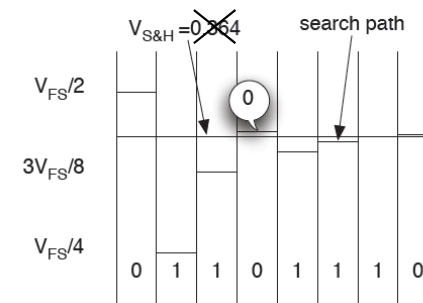


SA – algorithm



The name of the algorithm comes from the fact that the voltage from the DAC is an improving approx. of the voltage from the S&H – occasionally, the error can be larger than in the previous step (as from 4th to 5th bit below), but surely is not larger than successive divisions by 2 of the full-scale amplitude

Example below: search path for $V_{S\&H} = 0.4296875V_{FS}$



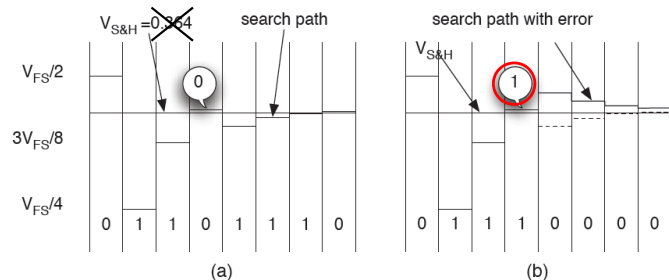
Algorithm with error



If there is an error in the bit evaluation, the error propagates along all successive steps – for instance, this may happen if V_{DAC} changes from a level well below $V_{S\&H}$ to a level just above $V_{S\&H}$ (4th clock period below) – if the comparator recovery from overdrive is not fast enough, an error may occur – in the case below, we end up with a conversion error of 2 LSBs.

Occurs typically at the beginning of the conversion, when overdrive is large

Error correction techniques expand the search range near the end, to accommodate for initial inaccuracies → however, extra clock cycles needed



Charge-redistribution SA-ADC

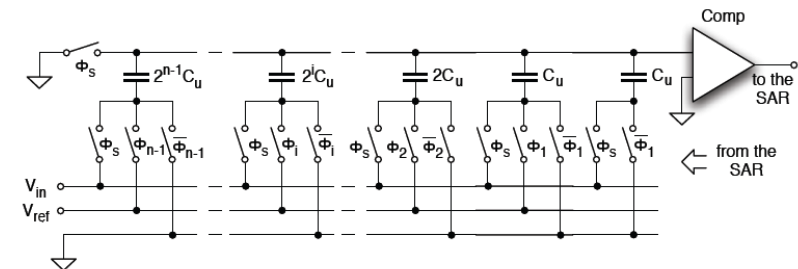


Charge sampled at the beginning of the conversion is redistributed on the capacitor array, to obtain a top-plate voltage close to zero at the end

Binary-weighted capacitances + a comparator → only one active-device block, which is not even particularly critical → very attractive for nanometer CMOS processes

During Φ_s , sampling → array connected between V_{in} and ground – total charge becomes:

$$C_{tot} = 2^n C_u V_{in}$$

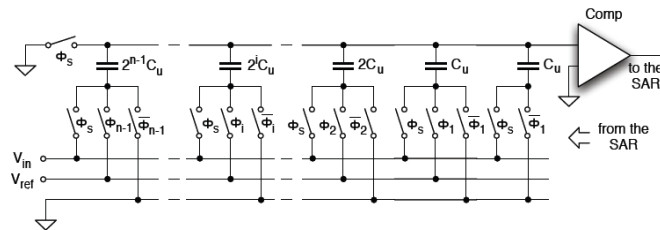


Charge-redistribution SA-ADC - conversion

MSB \rightarrow bottom plate of largest capacitance ($2^{n-1} C_u$) to V_{ref} , rest of the array to ground \rightarrow superposition yields the voltage on the top plate:

$$V_{comp}(1) = \frac{V_{ref}}{2} - V_{in}$$

This voltage is the difference between MSB and input \rightarrow only necessary to compare it to zero = ground, very convenient; if MSB=1, the connection of the largest cap to V_{ref} is kept during the 2nd comparison; otherwise it is restored to ground



SA-ADC conversion – II

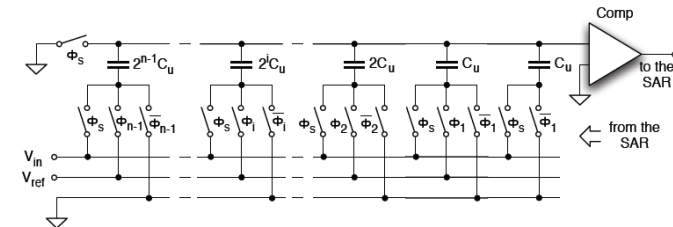
Thus, during the second comparison, the top plate voltage becomes

$$V_{comp}(2) = \frac{V_{ref}}{2} \cdot MSB + \frac{V_{ref}}{4} - V_{in}$$

which is used to find the 2nd bit; and so on for all bits

The parasitic capacitance at the top plate attenuates the generated voltage by a factor α :

$$\alpha = \frac{C_u 2^n}{C_u 2^n + C_p}$$



SA-ADC conversion – good features

α reduces the voltage value, but not its sign, which is the relevant info – this is a consequence of pre-charging the top-plate to zero – the top-plate voltage is zero at sampling, and almost zero at the end of the conversion

The input common-mode range of the comparator is zero without using op-amps or OTAs

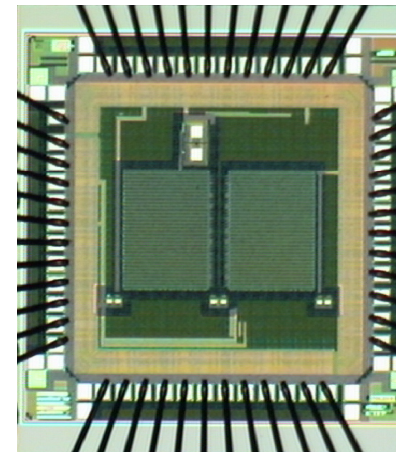
Only the comparator and charging/discharging the array determine the power consumption \rightarrow very power efficient (however, this does not take into account the generation of V_{REF})

Auto-zeroing of the comparator to avoid offset errors \rightarrow comparator is connected as a unity-gain buffer and used to pre-charge the top plate during sampling

Capacitive attenuation can be used to limit the capacitive spread in the array

Example of SA-ADC

A Programmable 10b up-to-6MS/s SA-ADC Featuring Constant FoM with On-Chip Reference Voltage Buffers



CMOS Technology	0.13 μm 1P6M
Core area [mm ²]	0.75
Power Supply [V]	1.2
Power consumption [mW]	3.2
Sampling frequency [MS/s]	5.5
INL/DNL [LSB]	0.6/0.55
ENOB [bits]	9.2
HD3 \approx IM3 [dBs-FS]	72
FoM [pJ/conversion]	1

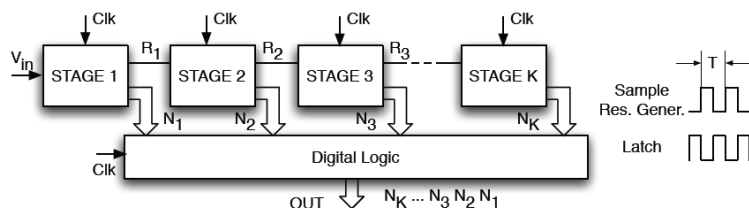
Pipeline ADC

Cascade of individual stages, each performing one of the elementary functions required by a sequential algorithm

The pipeline unwinds over space what would be performed over time by a sequential scheme

The simplest sequential scheme is the two-step algorithm, which uses two clock periods, one for MSBs and one for LSBs → the pipeline version delivers MSBs and LSBs in one clock period, in this way: the first stage yields the MSBs of the current sample, and the second stage yields the LSBs of the previous sample

In general, pipeline ADCs are multi-step (> 2)



Pipeline ADC - II

SA with pipeline → each stage of the pipeline generates 1 bit, plus the difference between input and internal DAC voltage – accuracy of analog signal must comply with the desired # of bits (each stage may deliver more than 1 bit, and/or different stages may deliver different # of bits)

Digital logic combines the bits coming from K stages → outputs at **full rate**, but with a delay (latency) of K+1 clock cycles → not a problem, unless the ADC is inside a feedback loop

S&H	S&H n	S&H n+1	S&H n+2				
Stage 1	b_9-b_8 n-1	b_8-b_8 n	b_7-b_8 n+1	b_6-b_8 n+2			
Stage 2		b_7-b_6 n-1	b_7-b_6 n	b_7-b_6 n+1	b_7-b_6 n+2		
Stage 3			b_5-b_4 n-1	b_5-b_4 n	b_5-b_4 n+1	b_5-b_4 n+2	
Stage 4				b_3-b_2 n-1	b_3-b_2 n	b_3-b_2 n+1	b_3-b_2 n+2
Stage 5					b_1-b_0 n-1	b_1-b_0 n	b_1-b_0 n+1
Digital					OUT n-1	OUT n	

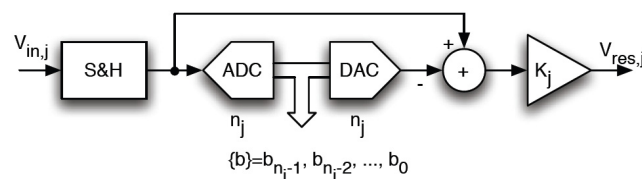
We see here a timing example for a 10-b, 5-stage, 2-b per stage pipeline ADC – the 6th clock cycle is used by the digital logic to combine the 10 bits and make them available

Generic pipeline stage

The subtraction of V_{DAC} from V_{in} gives the quantization error, which, after amplification, determines the new residue voltage

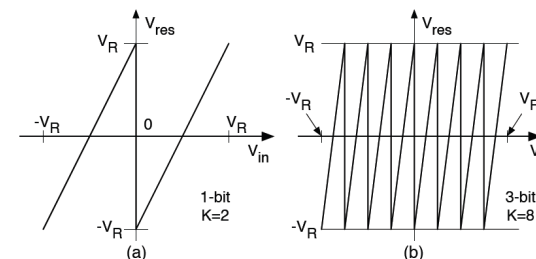
$$V_{res}(j) = K_j \{ V_{res}(j-1) - V_{DAC}(b_j) \}$$

The dynamic range of the residue equals that of the input if, for an n_j -bit DAC, the gain is 2^{n_j} → this is frequently used, as it allows the use of the same reference voltages in all stages



Residue generation

- input between $-V_R$ and V_R , 1 bit → the DAC subtracts $\frac{1}{2}V_R$ when the input is negative, and $\frac{1}{2}V_R$ when the input is positive; with $K=2$, the residue varies again between $-V_R$ and V_R
- 3 bits → the amplitude of the quantization error is at most $\pm V_R/8$, and $K=8$ makes the residue vary again between $-V_R$ and V_R

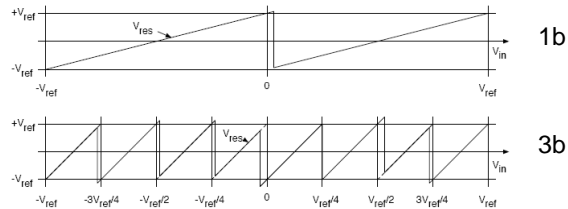


Accuracy requirements



Accuracy requirements are of course greater in the first few stages – most demanding is the input S&H

Non-idealities of ADC, DAC, and K cause limitations similar to those studied in the two-step ADC – threshold errors in the ADC cause the residue to be either greater or lower than full scale at the break points



However, the residue generator can still correctly provide the difference between the analog input and the quantized signal from the DAC – the error will be generated by the ADC in the next stage, since this ADC will not be able to correctly convert residues outside $\pm V_{ref}$ → this observation is the basis of the digital error correction technique to be treated next

Accuracy requirements – II



We observed that in the two-step ADC the DAC errors modified the residue over a whole LSB segment, impacting the INL → also in the pipeline ADC the accuracy of any DAC, referred to the input of the pipeline, must be better than the required INL, and lower than 1LSB to ensure monotonicity (after a few stages, DAC linearity is of minor concern)

If the interstage gain has an error, $G = 2^{n_j} (1 + \delta G)$, the slope of the residue is either increased or decreased, causing an error that is zero in the middle, and maximum at the endings of the segments – at the break points, the error invert its sign, giving rise to a step change

$$\Delta V = 2V_{ref} \delta G$$

which, referred to the input, must again comply with INL requirements and monotonicity

Digital error correction



We observed that the dynamic range of the residue can exceed the reference, but no error is made until the following ADC cannot convert correctly an out-of-range signal

Reduce the interstage gain? Difficult, as an attenuation different from 1/2 is difficult to account for

Add additional levels to the ADC of the stage? yes, because: 1) the redundant levels avoid generating out-of-range residues, and 2) provide info to the digital domain (hence the name of the technique) to fully compensate for the ADC error

Consider a 1-b DAC, which usually requires an ADC with 1 threshold – to ensure enough redundancy, at least 2 thresholds are needed

Since a 1-b ADC needs 1 threshold and a 2-b ADC needs 3, the use of 2 thresholds is referred to as a **1.5-b conversion** (very popular)

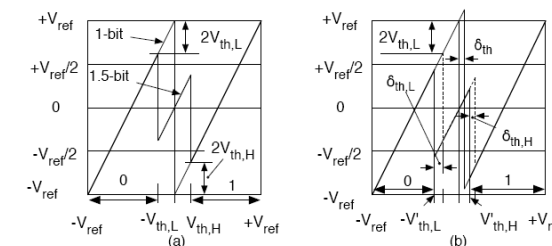
It is possible to apply the same approach to multi-bit DACs

Digital error correction – II



The input is divided into 3 regions: one below the lower threshold ($V_{th,L}$), one between the two thresholds, and one above the upper threshold ($V_{th,H}$) – if the separation of the two thresholds is large enough, a signal below $V_{th,L}$ is “certainly negative”, and above $V_{th,H}$ is “certainly positive” Uncertainty arises in the middle region → close to the zero crossing, an error may lead to a residue exceeding the limits

The residue generator adds $V_{ref}/2$ if the ADC provides a certain 0, and subtracts $V_{ref}/2$ if the ADC provides a certain 1 – but does not do anything in the uncertainty region → here the residue is the simple amplification by 2 of the input, shown below with correct (left) and incorrect thresholds

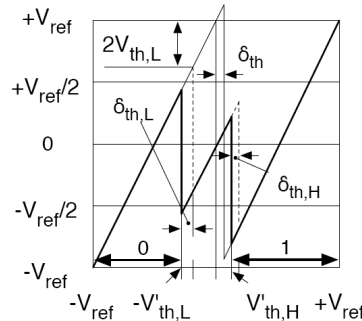


Digital error correction – III



Errors $\delta_{th,L}$, $\delta_{th,H}$ on the thresholds change the value of the residue at the break points, but the residue remains within $\pm V_{ref}$ if $\delta_{th,L}$ ($\delta_{th,H}$) is lower than $V_{th,L}$ ($V_{th,H}$) – in the case below, the input range for “certain” 0 diminishes, and increases for “certain” 1

The info provided by the 1.5-b converter is a flag in the LSB position, which is set to 1 when the uncertainty region is detected \rightarrow the whole code becomes therefore: 00 for certain 0; 10 for certain 1; and 01 for uncertainty.



Digital error correction – IV



The digital logic sums the outputs, taking into account the weight of each stage – the gain of each stage is 2 \rightarrow the weight of the uncertainty flag is equal to the MSB in the next stage (adders are needed)

In the example below, notice that the uncertainty in the 3rd stage is corrected by the 10 of the next two stages, while the 01 of the 4th and 5th stages are fixed to 0

The last stage does not have an extra threshold, as its uncertainty cannot be resolved by further comparisons

stage	1	2	3	4	...	n
time = k	x					
k+1		x				
k+2			x			
k+3				x		
...						
k+n-1						x
out						

(a)

stage	1	2	3	4	...	n-1	n
time = k	xx						
k+1		xx					
k+2			xx				
...							
k+n-2					xx		
k+n-1						x	
out							

(b)

stage	1	2	3	4	5	6
time = k	00					
k+1		10				
k+2			01			
k+3				01		
...						
k+4					10	
k+5						1
out	0	1	1	0	0	1

(c)

Digital error correction – example



Assume $\pm V_{ref} = \pm 1$, and $V_{in} = 2V_{ref} (1/4 + 1/8 + 1/64 + 10^{-3}) - V_{ref} = -0.21675 = 011001$

If $V_{th,L} = -0.25$, $V_{th,H} = 0.25$, we have the following residue sequence: -0.21675, -0.4335, 0.133, 0.266, -0.468, 0.064, corresponding to the binary outputs: 01, 00, 01, 10, 00, 1 \rightarrow 011001, as expected

If now shift on thresholds: $V_{th,L} = -0.30$, $V_{th,H} = 0.10$, the sequence becomes -0.21675, -0.4335, 0.133, -0.734, -0.468, 0.064, corresponding to the binary outputs: 01, 00, 10, 00, 00, 1 \rightarrow still 011001 !

With thresholds $V_{th,L} = -0.10$, $V_{th,H} = 0.30$, the sequence becomes -0.21675, 0.5665, 0.133, 0.266, 0.532, 0.064, corresponding to the binary outputs: 00, 10, 01, 01, 10, 1 \rightarrow again 011001 !

Dynamic performances



Depend on slew-rate and bandwidth of S&H and residue generator – a step with amplitude $\overline{V_{out}}$ becomes a ramp during slewing, and turns into an exponential when feedback takes over – the equations are (as already derived in an earlier lecture)

$$V_{out}(t) = SR \cdot t \quad t < t_{slew}$$

$$V_{out}(t) = \overline{V_{out}} - \Delta V \cdot e^{-(t-t_{slew})/\tau} \quad t > t_{slew}$$

$$\Delta V = SR \cdot \tau; \quad t_{slew} = \overline{V_{out}} / SR - \tau; \quad \tau = 1 / \beta \omega_T$$

feedback factor

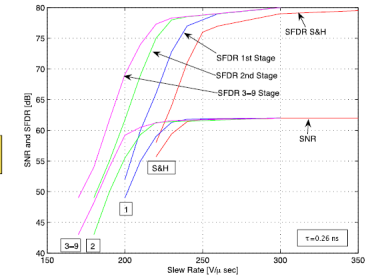
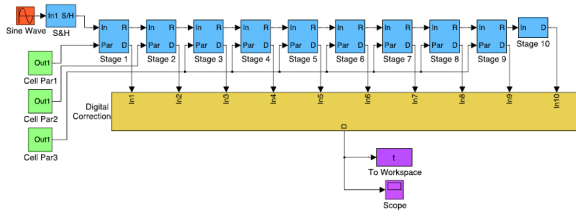
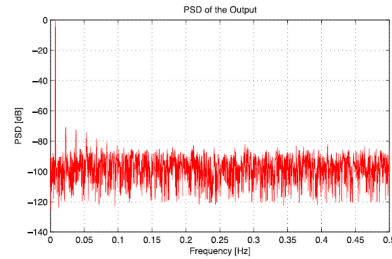
The next stage S&H samples after $T_S/2$, and the (non-linear) settling error becomes

$$\Delta V_{err} = \Delta V \cdot e^{-(T_S/2 - t_{slew})/\tau}$$

It is important to have a small error in the first stages, but in later stages is less important, because the input-referred error is divided by the gain of all preceding stages \rightarrow op-amps in later stages do not need to be as high-performing \rightarrow it is possible to save power

Behavioral simulations

Much faster than transistor-level simulations, give a good starting point for the design – here, the achievable SFDR and SNR vs. the slew-rate of the key blocks are estimated – the different impact of the different blocks is also very easy to determine



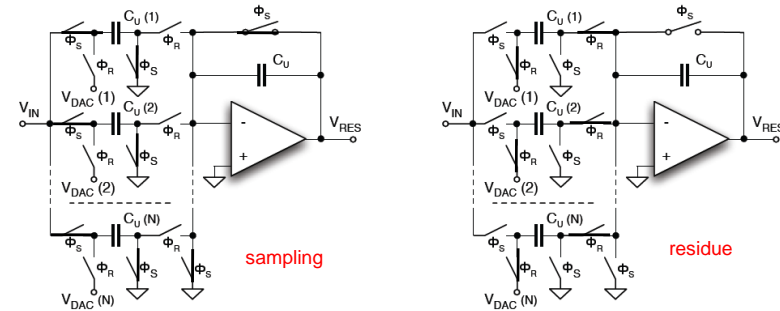
Residue generator

Capacitors are used for both sampling and D/A conversion; the residue is

$$V_{res} = NV_{in} - \sum_{i=1}^N V_{DAC}(i)$$

where the DAC control bits are thermometric from the ADC

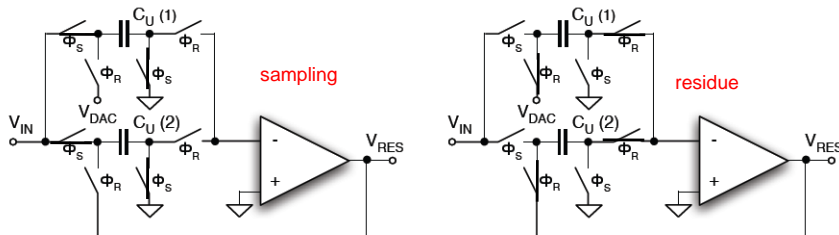
An amplification of 2^n gives rise to a feedback factor $\beta = 1/(2^n + 1) \rightarrow$ to keep the same τ , the unity-gain frequency of the op-amp must grow as $2^n \rightarrow$ more than 3 bits per stage not suitable for high conversion rates



Residue generator – II

With 1 bit $\rightarrow \beta=1/3$ (2 caps are needed at the input to implement the gain x2) – speed can be improved with the flip-around circuit shown below, where $\beta=1/2$, since input and feedback capacitance are equal (however, is the op-amp really open-loop during sampling?)

If $C_U(1)=C_U(2)$, the input voltage is doubled (one cap changes polarity compared to the other), but the gain for V_{DAC} is only -1 \rightarrow it is necessary to double the value of the references.



Cyclic (or algorithmic) converter

Uses the same cell for converting one bit per clock cycle $\rightarrow n+1$ clock periods to convert n bits; the voltage at the output of OTA 1 is the same as the residue of the first stage of a 1-b per stage pipeline:

$$V_{OTA1} = 2V_{in} - V_{REF} = 2\left(V_{in} - \frac{V_{REF}}{2}\right)$$

