Advanced AD/DA converters

ΔΣ Toolbox; Example of DT ΔΣ Modulator

Pietro Andreani
Dept. of Electrical and Information Technology
Lund University, Sweden

Overview

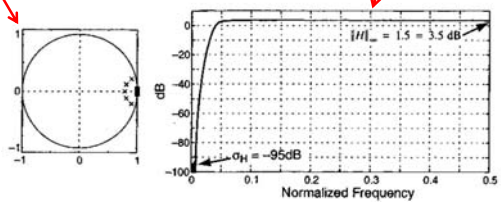
- Delta-Sigma Toolbox – some of the key functions
- 2nd-order DT modulator

synthesizeNTF

synthesizeNTF → finds an NTF with specified order and out-of-band gain (H_inf), having: 1) optimized zeros (if desired), and 2) poles of a maximally-flat all-pole transfer function

```
order = 5;
OSR = 64;
opt = 1;                    % optimized zeros
H_inf = 1.5;                % defaults to 1.5
H = synthesizeNTF(order, OSR, opt, H_inf);
plotPZ(H);
f = linspace(0, 0.5, 1000);
z = exp(2i*pi*f);
plot(f, dbv(evalTF(H,z)));
sigma_H = dbv(rmsGain(H, 0, 0.5/OSR));
```

N	zero locations, normalized to ω_s	SQNR improvement
1	0	0 dB
2	$\pm 1/(\sqrt{3})$	3.5 dB
3	$0, \pm\sqrt{3/5}$	8 dB
4	$\pm\sqrt{3/7 \pm \sqrt{(3/7)^2 - 3/35}}$	13 dB
5	$0, \pm\sqrt{5/9 \pm \sqrt{(5/9)^2 - 5/21}}$	18 dB
6	±0.23862, ±0.66121, ±0.93247	23 dB
7	0, ±0.40585, ±0.74153, ±0.94911	28 dB
8	±0.18343, ±0.52553, ±0.79667, ±0.96029	34 dB



Assuming $\sigma_e^2 = 1/3 = -4.8\text{dB}$, we obtain:

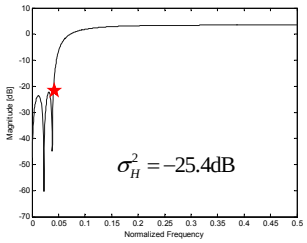
$$\sigma_q^2 = \sigma_{e,in-band}^2 \sigma_H^2$$

$$= \frac{\sigma_e^2}{OSR} \sigma_H^2 = -118\text{dB}$$

synthesizeNTF – II

This choice of poles is convenient → when CIFB or CRFB topologies are used with a single feed-in, the STF has the same frequency response as the all-pole transfer function created by the poles of the NTF

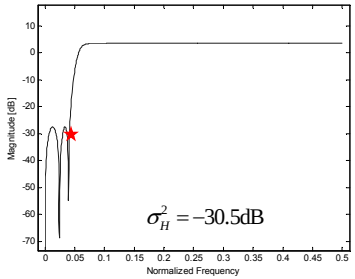
The NTF performance is primarily determined by zero locations and out-of-band gain, while pole locations are of secondary importance (since the denominator is basically constant in-band) – this is of course less true if the OSR is low, since in this case the -3dB cutoff frequency of the denominator gets closer to the passband (e.g., see the plot below for a 5th-order modulator with $\|H\|_\infty = 1.5$ and OSR=12)



synthesizeNTF – III

Another limitation is when $\|H\|_\infty$ is close to unity and zeros are optimized. With a low value of $\|H\|_\infty$ the poles of H converge to z=1. If all zeros of H also are at z=1, $\|H\|_\infty$ approaches 1 and there is no problem; however, if the zeros are optimal, $\|H\|_\infty$ does not converge to 1 any more – this problems are due to the fact that poles and zeros are optimized separately (zeros taken from previous table), and not taking each other into account

If $\|H\|_\infty$ is close to 1 or OSR is low → use *synthesizeChebyshevNTF* → still not optimal, but better than the standard *synthesizeNTF*



Some theory

Take P(z) = n[th]-order polynomial, with $\left|P\left(e^{j\omega}\right)\right|$ maximally flat around ω=0

P(z) will be the denominator of the NTF

The coefficients of P(z) are real → $\left|P\left(e^{j\omega}\right)\right|^2 = P\left(e^{j\omega}\right)\left[P\left(e^{j\omega}\right)\right]^* = P(z)P\left(\frac{1}{z}\right)_{z=e^{j\omega}}$

Maximally flat around z=1 →

$$P(z)P\left(\frac{1}{z}\right) = P^2(1) + a(z-1)^n\left(\frac{1}{z}-1\right)^n$$

If a=0 → P is constant (very flat!); a larger a yields a lowpass function with increasing sharpness; a must be positive, otherwise P(z) has a decreasing magnitude when moving away from z=1 (the opposite of what we want); taking P(1)=1 we obtain

$$P(z)P\left(\frac{1}{z}\right) = 1 + a\frac{(z-1)^{2n}}{(-z)^n} = \frac{a(z-1)^{2n} + (-z)^n}{(-z)^n}$$

Some theory – II

The roots of $a(z-1)^{2n} + (-z)^n = 0$ are the poles (and their inverse) of the desired NTF →

$$a(z-1)^{2n} = -(-z)^n \quad \rightarrow \quad e^{j2\pi k}a(z-1)^{2n} = e^{-j\pi}(-z)^n \quad \rightarrow \quad e^{\frac{j(2k+1)}{n}\pi}a^{\frac{1}{n}}(z-1)^2 = -z$$

Thus, the roots are given by the *n* complex quadratic equations

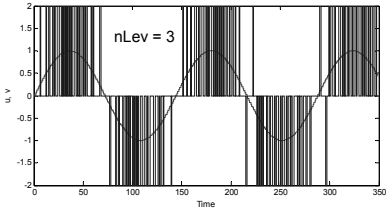
$$z^2 + b_k z + 1 = 0, \qquad b_k = \frac{e^{\frac{j(2k+1)}{n}\pi}}{a^{\frac{1}{n}}} - 2, \qquad k = 0...n-1$$

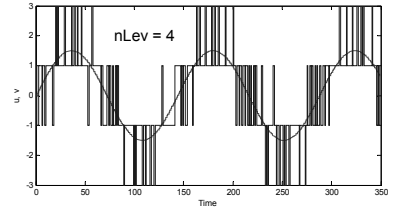
The product of the two roots of each equation is 1 → one root is inside the unit circle, the other outside → collecting all roots inside the circle yields the poles of the desired NTF

simulateDSM

simulateDSM → time-domain simulations of the modulator found with *synthesizeNTF*, assuming that STF is unity

```
OSR = 64;
nLev = 3;      % number of levels in the quantizer
Nfft = 2^13;
tone_bin = 57;
t = [0:Nfft-1];
u = 0.5*(nLev-1)*sin(2*pi*tone_bin/Nfft*t);
                % nLev-1 = max signal
                % 0.5*(nLev-1) = -6dBFS
v = simulateDSM(u, H, nLev);
n = 1:350;
stairs(t(n), u(n), 'r');
hold on;
stairs(t(n), v(n), 'b');
```



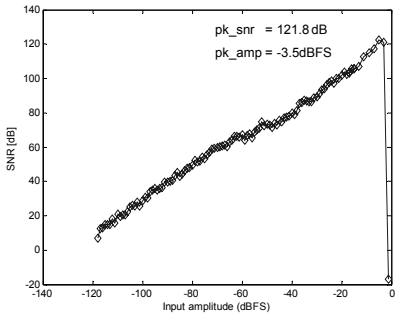


simulateSNR

calculateSNR;

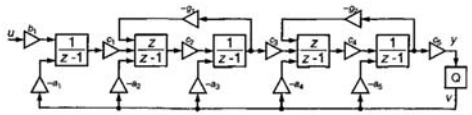
simulateSNR → the amplitude of the input signal is swept (however, this function does seem to have problems!)

```
OSR = 64;
nLev = 3;
amp = [-130:5:-20  -17:2:-1];
snr = simulateSNR(H, OSR, amp, [], nLev);
plot(amp, snr, '-b', amp, snr, 'db');
[pk_snr pk_amp] = peakSNR(snr, amp)
```

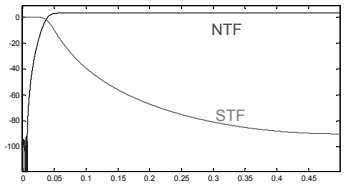


realizeNTF and associated functions

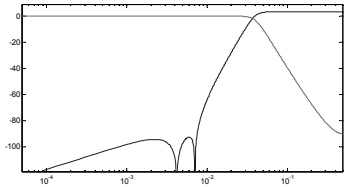
The synthesized NTF (and STF) are mapped here to a CRFB modulator with *realizeNTF* (*synthesizeNTF* returns STF=1 – setting all b_i except b_1 to zero, we obtain a maximally-flat all-pole STF, see here below)



```
H = synthesizeNTF(5, 64, 1);
form = 'CRFB';
[a, g, b, c] = realizeNTF(H, form);
b(2:end)=0;   % maximally flat STF
  % stuffABCD → to calculate NTF/STF from
  % generic a-b-g-c coefficients and given topology
ABCD = stuffABCD(a, g, b, c, form);
[Ha Ga] = calculateTF(ABCD);
  % Ha = NTF, Ga = STF;
f = linspace(0, 0.5, 10000);
z = exp(2i*pi*f);
magHa = dbv(evalTF(Ha,z));
magGa = dbv(evalTF(Ga,z));
plot(f, magHa, 'b', f, magGa, 'm', 'Linewidth', 1)
```



linear axis

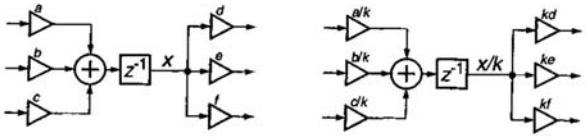


log axis

Scaling of dynamic range

realizeNTF returns unscaled coefficients → NTF and STF are ok, but there is no control of the internal states (i.e. integrator outputs) → dynamic range scaling is a must – this is an issue common to all active filter implementations!

Scaling is accomplished by dividing the admittance of all input branches of a given integrator by a factor *k*, and multiplying with the same factor the admittance of all output branches of the same integrator – in this way, the rest of the circuit is unaffected, and so are the transfer functions



scaleABCD

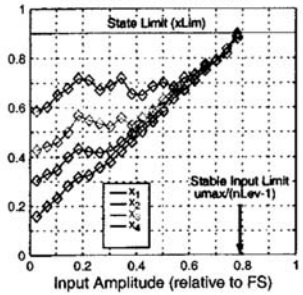
scaleABCD → 1) determines maximum stable input amplitude (*umax*) and maximum value for each modulator state for inputs up to *umax*; 2) dynamic range scaling is applied → maximum value of each state does not exceed the specified *xLim* (remember: 0dBFS = nLev – 1)

mapABCD (inverse of *stuffABCD*) → maps the results in terms of coefficients for the desired topology

Scaled Coefficients:

i	a_i	g_i	b_i	c_i
1	0.0383	0.0034	0.0383	0.1473
2	0.0748	0.0045	0	0.2045
3	0.0969	–	0	0.4007
4	0.1754	–	0	0.4385
5	0.1709	–	0	3.2533

Observed State Maxima



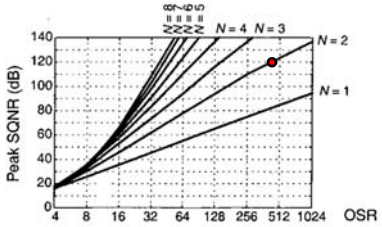

```
nLev = 3; xLim = 0.9; f0 = 0;
[ABCDs umax] = scaleABCD(ABCD, nLev, f0, xLim);
[a g b c] = mapABCD(ABCDs, form);
```

Example of 2nd-order modulator

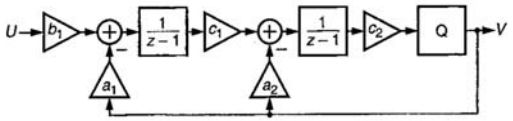
Low-speed modulator, e.g. for on-chip calibration engine; f_b=1kHz and f_s=1MHz → OSR=500, 1-bit DAC → SQNR ≈ 120dB → SNR in excess of 100dB possible

We assume that VDD is used as reference voltage (disregard supply noise)

Parameter	Symbol	Value	Units
Bandwidth	f_B	~1	kHz
Sampling Frequency	f_s	1	MHz
Signal-to-Noise Ratio	SNR	100	dB
Supply Voltage	VDD	3	V



We start selecting the standard CIFB topology



Code

```
H = synthesizeNTF(2, 500, 0, 2);          %out-of-band NTF peak gain = 2
form = 'CIFB';
[a, g, b, c] = realizeNTF(H, form);
b(2:end)=0;
ABCD = stuffABCD(a, g, b, c, form);
[ABCDs umax] = scaleABCD(ABCD);    % default:  xLim=1, nLev=2
[a g b c] = mapABCD(ABCDs, form);
```

a = 0.2653, 0.2212

b = 0.2653, 0, 0

c = 0.3185, 5.5874 (c_2 is not important in a single-bit quantizer)

Rounding →

a_1 = 1/4, a_2 = 1/4, b_1 = 1/4, c_1 = 1/3

Effect of rounding: peak NTF=2.25, peak SQNR=115dB

From simulations, effective quantizer gain (including c_2) is approx. 16/3

Code and signal denormalization

```
a = [1/4 1/4];
b = [1/4 0 0];
c = [1/3 1];
ABCD = stuffABCD(a, g, b, c, form);
k = 16/3;
NTF = calculateTF(ABCD, k);
```

$$NTF(z) = \left(\frac{z-1}{z-1/3}\right)^2$$

umax = 0.966 (normalized to V_{ref}=V_{DD}) → almost rail-to-rail input dynamic range

The toolbox assumes that the input of a binary modulator is between -1 and +1; the same for the integrator states after dynamic range scaling – everything is of course unit-less

In the following example, the full-scale input is $3V_{pp}$, while the full-scale in the toolbox is 2_{pp}. Let us assume that the amplifier supports a differential swing with the same numerical range as the toolbox, i.e. $2V_{pp}$, and that the digital signal v_d in the circuit is interpreted as either 0 or 1 (corresponding to open or closed switch)

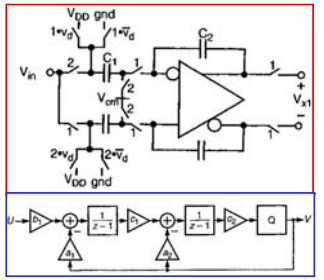
Equations

The relationship between state variables (u, x_1, x_2, v) and circuit variables (v_{in}, v_{x1}, v_{x2}, v_d) becomes:

$$v_{in} = u \cdot \text{scale factor} + v_{cm} = \frac{3V}{2}u + 1.5\text{V} \Rightarrow \begin{matrix}\text{state} & \text{circuit}\\ [1 \quad -1] & \rightarrow [3 \quad 0]\end{matrix}$$

$$v_{x1} = x_1 \cdot 1\text{V} \quad \Rightarrow \quad [1 \quad -1] \rightarrow [1 \quad -1]$$

$$v_d = \frac{v+1}{2}\text{V} \quad \Rightarrow \quad [1 \quad -1] \rightarrow [1 \quad 0]$$



Thus, the DT circuit is retrieved from the normalized model by performing the following substitutions:

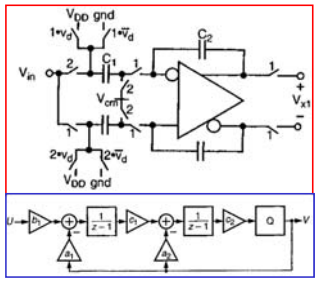
$$u \rightarrow \frac{v_{in} - 1.5}{1.5} \qquad x_1 \rightarrow v_{x1} \qquad v \rightarrow 2v_d - 1$$

Equations

$$u \rightarrow \frac{v_{in}-1.5}{1.5}$$

$$x_1 \rightarrow v_{x1}$$

$$v \rightarrow 2v_d - 1$$



Therefore, the 1st integrator equation

becomes

$$x_1(n+1) = x_1(n) + b_1 u(n) - a_1 v(n) = x_1(n) + \frac{1}{4}u(n) - \frac{1}{4}v(n)$$

$$v_{x1}(n+1) = v_{x1}(n) + b_1 \frac{v_{in}(n)-1.5}{1.5} - a_1\left[2v_d(n)-1\right] = v_{x1}(n) + \frac{v_{in}(n)}{6} - \frac{1}{2}v_d(n)$$

On the other hand, the SC circuit above approx. yields (more on this soon)

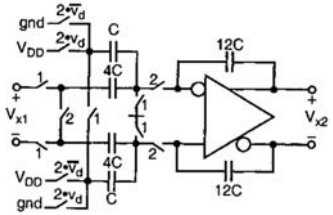
$$v_{x1}(n+1) = v_{x1}(n) + 2\frac{C_1}{C_2}v_{in}(n) - 2\frac{C_1 V_{dd}}{C_2}v_d(n)$$

Capacitance ratios

Since V_{dd}=3V, we obtain the capacitance ratio $\frac{C_1}{C_2} = \frac{1}{12}$

Notice that input capacitor and feedback capacitor are shared, i.e. are the same component

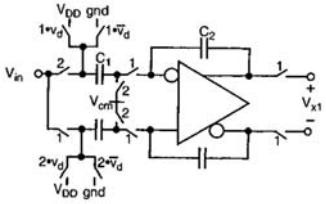
This is not true for the second integrator – the same procedure yields the ratios as in the figure below



Signal – first integrator

During ph1, V_{in} is transferred, inverted and without delay, to the bottom output through the bottom C_1

During ph2, V_{in} is loaded onto the top C_1, and then transferred, non-inverted, to the top output through the top C_1 during next ph1 (i.e., delayed by 1 clock cycle)



The transfer function from input to differential output becomes

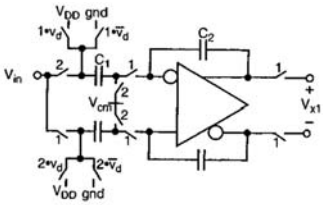
$$\left.\begin{aligned} v_{x1,top}(n) &= v_{x1,top}(n-1) + \frac{C_2}{C_1}v_{in}(n-1) \;\rightarrow\; V_{x1,top}\left(1-z^{-1}\right) = \frac{C_2}{C_1}z^{-1}V_{in} \\ v_{x1,bot}(n) &= v_{x1,bot}(n-1) - \frac{C_2}{C_1}v_{in}(n) \;\rightarrow\; V_{x1,bot}\left(1-z^{-1}\right) = -\frac{C_2}{C_1}V_{in} \end{aligned}\right\} \rightarrow \frac{V_{x1,diff}}{V_{in}} = \frac{C_2}{C_1}\frac{1+z^{-1}}{1-z^{-1}}$$

This is the *bilinear (trapezoidal)* integration method mapping CT to DT → half delaying (Euler forward), half non-delaying (Euler backward) → better than either Euler methods

Noise – I

Assuming the opamp noise negligible, the input-referred noise of the first integrator is

$$v_{n1}^2 = \frac{k_B T}{C_1}$$



Why? During ph1, a noise voltage $v_{n1,a}$ is loaded on top C_1, while a noise voltage of $v_{n1,b}$ and the signal v_{in} are loaded on bottom C_1; during ph2 a noise voltage $v_{n1,c}$ and the signal v_{in} are loaded on top C_1, and the noise voltage $v_{n1,d}$ is loaded on bottom C_1. Therefore, since all noise voltages are uncorrelated, at the integrator output we obtain:

$$v_x^2 = \left(v_{in}+v_{in}\right)^2 + v_{n1,a}^2 + v_{n1,b}^2 + v_{n1,c}^2 + v_{n1,d}^2 = 4\left(v_{in}^2 + v_{n1}^2\right)$$

Thus, both signal and kT/C noise are multiplied by the same factor, and therefore the input referred noise is simply v_{n1}

Noise – II

The kT/C noise is white between DC and Nyquist → its in-band power is therefore

$$v_{n1}'^2 = \frac{v_{n1}^2}{OSR}$$

In order to achieve an SNR of 100dB with a full-scale sine input, we must have

$$10^{10} = \frac{(1.5V)^2/2}{v_{n1}'^2} \quad \rightarrow \quad v_{n1}'^2 \approx (10\mu V)^2$$

If we allocate all noise to the first integrator, and assuming T=300K, we obtain

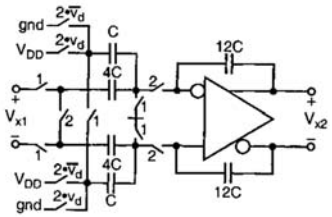
$$(10\mu V)^2 = \frac{k_B T}{C_1 \cdot OSR} \quad \rightarrow \quad C_1 = \frac{1.38\cdot 10^{-23}\cdot 300}{(10\mu V)^2 \cdot 500} \approx 74\,fF$$

from which C_2=0.88pF (in reality, we need some margin on these values!)

Noise – III

Assuming again the opamp noise negligible, the input-referred noise of the second integrator is

$$v_{n1}^2 = 2\left(2\frac{k_B T}{4C} + 2\frac{k_B T}{C}\right) = 5\frac{k_B T}{C}$$



This noise is shaped by the inverse of the transfer function of the first integrator →

$$H_1(z) = \frac{C_2}{C_1}\frac{1+z^{-1}}{1-z^{-1}} \approx \frac{2C_1/C_2}{1-z^{-1}}$$

Thus, the in-band noise becomes $v_{n2}'^2 = \frac{\pi^2}{3OSR^3}\left(\frac{C_2}{C_1}\right)^2 v_{n2}^2 \approx 10^{-6} v_{n2}^2$

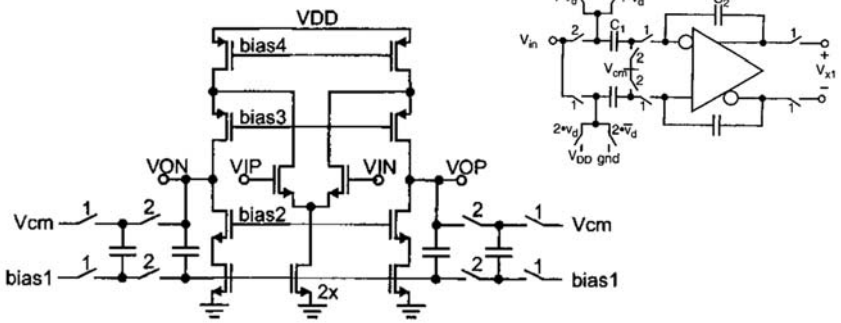
Thus, totally negligible even if we choose a very small C capacitor, e.g. C = 20fF → in this case, the minimum value for C is dictated by process limitation rather than noise considerations

Operational amplifier

Classical folded-cascode topology; with $2I_B$ current in the input pair, the slew current available at each output is I_B; the largest quantity that needs to be transferred from C_1 to C_2 is C_1V_{DD} → allocating ¼ of ½ clock period for slewing, we obtain

$$I_B = \frac{C_1 V_{DD}}{0.25\cdot 0.5\cdot T_{clk}} = 8 f_{clk} C_1 V_{DD} = 8\cdot 1M\cdot 83f\cdot 3 \approx 2\mu A$$

Thus, 8μA are enough for the whole amplifier



Operational amplifier – II

Bandwidth estimation → if we allocate 10 time constants τ of linear settling in the remaining ¾ of ½ clock period → $\delta_{err} = e^{-10} \approx -87dB$ → since this is the error from the onset of the linear settling, i.e. it is only a (small) fraction of the whole step, 100dB of SNR are easily obtainable

We have seen that $\tau \approx \frac{C_L}{\beta g_m}$, where C_L is the total capacitance loading the opamp output, g_m is the transconductance of the input transistor, and β is the feedback factor →

$$\beta = \frac{C_2}{C_1 + C_2}, \qquad C_L \approx \frac{C_1 C_2}{C_1 + C_2} \quad \rightarrow \quad \tau \approx \frac{C_1}{g_m} = \frac{1}{10} 0.5\cdot 0.75\cdot T_{clk} \quad \rightarrow g_m = 2.3\mu A/V$$

DC gain

We know that the NTF begins to degrade if its zeros are moved inside the unit circle by approx. π/OSR ; the finite gain of the opamp shifts the pole of the first integrator by $C_1/(AC_2)$ → it would seem that the following DC gain A would be adequate:

$$A=\frac{OSR\cdot C_1}{\pi C_2}\approx 13$$

This, however, comes from a linear analysis, and neglects non-linear errors coming from slewing and non-linear DC gain (if A is linear, a finite A does not cause distortion; if there is no slewing, settling errors due to finite bandwidth do not cause distortion)

Upper bound for required DC gain: the settled voltage at the opamp input (i.e. in series with the input signal) is V_{out}/A ($V_{out}/2A$ in the double-sampling of the first integrator): this voltage has a signal component (which is ok), broadband noise, and distortion: if we assume that it consists of only distortion (which is highly pessimistic), the requirement that distortion stay below -100dBFS yields

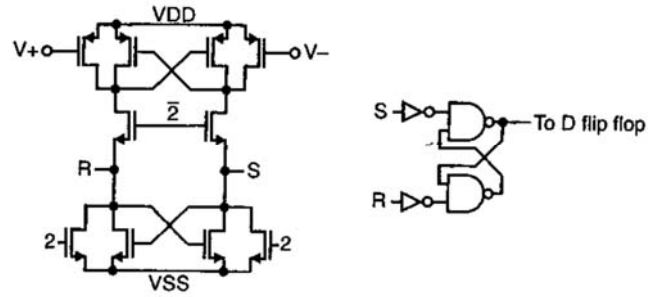
$$\frac{V_{out}}{2A}<10^{-5}V_{in} \quad\rightarrow\quad \frac{1}{2A}<10^{-5}\cdot 1.5 \quad\rightarrow\quad A>90dB$$

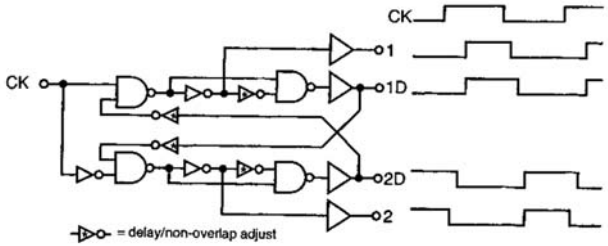
DC gain – II

In practice, A=60dB (!) should be adequate (from experience), but this has to be checked with extensive simulations using the real non-linear model of the opamp

Second integrator: we have seen that its noise requirements are much relaxed → the same is true with respect to DC gain and slew-rate → this amplifier can be implemented as a scaled-down version of the first amplifier, e.g. by a factor 4 (or higher, but the area and power savings decrease rapidly)

Latched comparator and clock generator



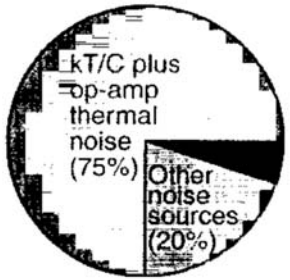


Noise budget

Important to find a good balance between all noise source → the various noise sources are scaled in a way to give the most economical implementation → e.g. if q-noise is allocated 90% of the noise budget, then the capacitor sizes that should satisfy the remaining 10% kT/C noise may be quite large → excessive area and power consumption

Furthermore, a large q-noise is not desirable → q-noise is not really random → may compromise performance in e.g. hi-fi audio systems, since the human ear can detect tones that are 20dB below the total noise level!

A reasonable noise budget is the one on the right



More on noise

When OSR is very large, the first integrator dominates the thermal noise budget – this is not true for low OSR (wide bandwidth) because the integrator gain is low at the high end of the signal band, which means that noise is not much attenuated