David Goldman

EE 515 – CMOS Mixed-Signal IC Design

Problem 31.1

Question: Develop an expression for the effective number of bits in terms of the measured signal-to-noise ratio if the input sinewave has a peak amplitude of 40% of (V_{REF+} - V_{REF-}).

Answer: Eq. (31.1) is the ratio in decibels of RMS input signal to the RMS quantization error (Eqs. (30.30) and (30.32)). Eq. (31.2) is developed in Chapter 30 (Eq. (30.23)). These equations are the starting point and are reiterated below

$$SNR_{ideal} = 20 \cdot \log \frac{V_p / \sqrt{2}}{V_{LSB} / \sqrt{12}}$$

$$V_{LSB} = \frac{V_{REF+} - V_{REF-}}{2^N}$$

The following relationship is stated in the question

$$V_p = (0.40) \cdot (V_{REF+} - V_{REF-})$$

or

$$(V_{REF+} - V_{REF-}) = \frac{5 \cdot V_p}{2}$$

V_{LSB} can now be written

$$V_{LSB} = \frac{5 \cdot V_P}{2 \cdot 2^N}$$

and the ideal SNR can be written as

$$SNR_{ideal}(dB) = 20 \cdot \log \frac{V_p / \sqrt{2}}{V_{LSB} / \sqrt{12}} = 20 \cdot \log \frac{V_p / \sqrt{2}}{\left(\frac{5 \cdot V_P}{2 \cdot 2^N} \right) / \sqrt{12}}$$

$$= 20 \cdot \log \frac{2^N \cdot 2\sqrt{12} \cdot V_P}{5\sqrt{2} \cdot V_P} = 20 \cdot \log \frac{2^N \cdot 4\sqrt{3}}{5\sqrt{2}}$$

$$= 20 \cdot \left(\log(2^N) + \log\left(\frac{4\sqrt{3}}{5\sqrt{2}} \right) \right) = 20N \cdot \log(2) + 20 \cdot \log\left(\frac{4\sqrt{3}}{5\sqrt{2}} \right)$$

$$= 6.02N - 0.177$$

(Note: when comparing the previous equation to Eq. (31.4), it is shown that as the number of bits increases the effect of a diminished input signal is increasingly less important in determining the SNR_{ideal}.)

The same equation can be converted from an ideal equation to a real equation by changing the variable names – ideal to measured and bits to effective bits

$$SNR_{meas} = 6.02N_{eff} - 0.177$$

Solving for the number of bits, a final answer is obtained

$$N_{eff} = \frac{SNR_{meas} + 0.177}{6.02}$$

So, by limiting the input to less than the full voltage range, the number of effective bits is increased to sample a signal with the same SNR. Conversely, if the input signal is decreased and the same number of sampling bits is used, the SNR will decrease.

Gexin

31.2: Determine a data conversion system's SNR if the measured $V_{Qe,RMS}$ is 1 mV and the maximum peak-to-peak amplitude of an input sinewave is 1V.

Solution for 31.2:

The RMS value for the given sinewave signal is $\frac{1}{2}\cdot\frac{1}{\sqrt{2}}V$.

So the SNR value for this data conversion system is:

$$SNR_{meas} = \frac{0.5/\sqrt{2}}{1mV} = 353.5 = 51 \text{ dB}$$

Brandon Roth
brandonroth@micron.com

31.3 Repeat Ex. 31.2 if the sampling frequency is increased to 200MHz. Does the SNR change?

From example 31.2, using an ideal 8-bit ADC and DAC find the SNR using SPICE. The input to the ADC is a 25MHz sine wave with a peak amplitude of 0.75V centered on a DC voltage of 0.75V.

$$V_{in} = 0.75 + 0.75\sin 2\pi(25MHz)t$$

First lets find the SNR_{ideal} value using hand calculations.

From Eq. 31.4

$$SNR_{ideal} = 6.02N + 1.76(dB) = 6.02(8) + 1.76 = 49.92dB \approx 50dB$$

Using the netlist from example 31.2, the following changes were made:

1. The Vclock pulse period and width were decreased to 5.0ns and 2.4ns respectively to increase the frequency to 200MHz.
2. The only spectrum components of the FFT that should be ignored are DC (0), 25MHz (the input signal), and 175MHz (were $175=f_s-f_{in}$, the first alias component). The range of the FFT was left at 200MHz so no further alias signals need to be excluded/ignored.
3. Finally the simulation stepsize had to be decreased from 2ns to 0.5ns. From Chapter 30

   Stepsize = 10% of T_s = 10%(5ns) = 0.5ns

   Without using the smaller stepsize the simulation returns 60mV for the total noise.

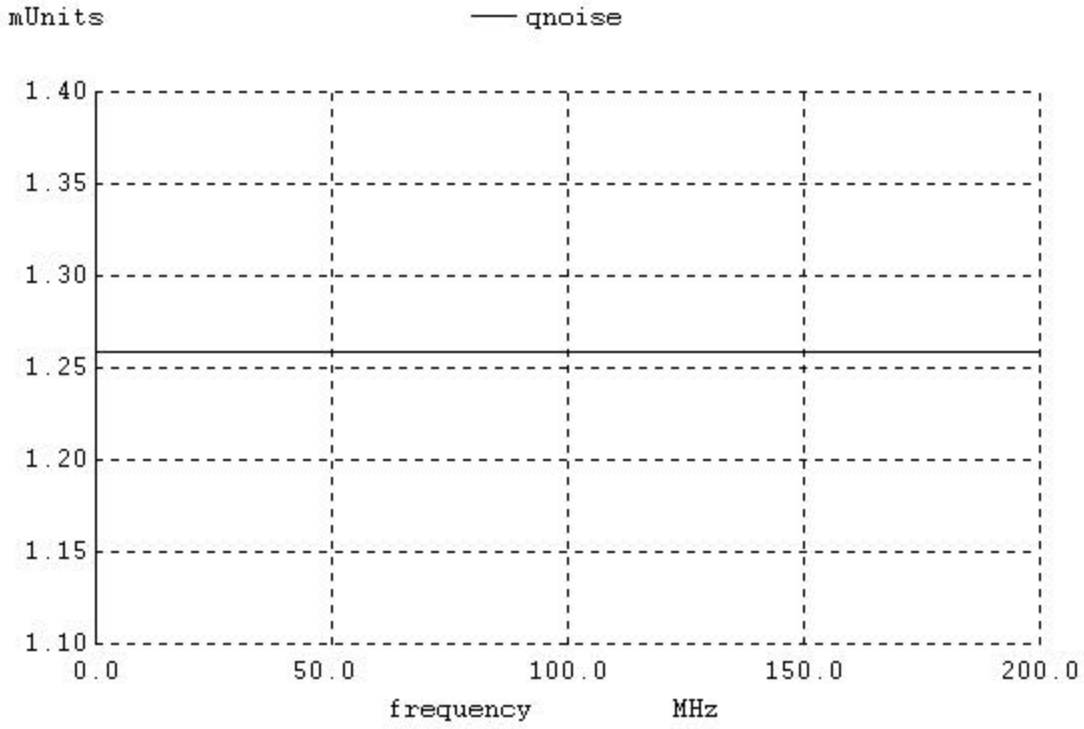
Below is the netlist with the changes described above,

```
* Problem 31.3 CMOS: Mixed-Signal Circuit Design *
.tran .5n 2000n 0 .5n UIC
.save Vout

*WinSPICE command scripts
*#destroy all
*#run
*#plot Vin Vout xlimit 0 100n
*#linearize Vout
*#spec 0 200MEG 1MEG Vout
*#plot db(vout)
*#let m=mag(vout)
*#let m[0]=0
*#let m[25]=0
*#let m[175]=0
*#let qnoise=0.707*sqrt(mean(m*m)*length(m))
*#plot qnoise

VDD VDD 0 DC 1.5
VREFP VREFP 0 DC 1.5
VREFM VREFM 0 DC 0
```

```
Vin    Vin   0 DC 0  Sin   0.75 0.75  25MEG
Vclock clock 0 DC 0  Pulse 0    1.5   0    200p  200p  2.4n 5n

X1 VDD VREFP VREFM Vin B7 B6 B5 B4 B3 B2 B1 B0 clock ADC8bit
X2 VDD VREFP VREFM Vout B7 B6 B5 B4 B3 B2 B1 B0 DAC8bit

*** Start Ideal DAC Subcircuit ******************************
.subckt DAC8bit VDD VREFP VREFM Vout B7 B6 B5 B4 B3 B2 B1 B0
*Generate Logic switching point, or trip, voltage
R1 VDD trip 100MEG
R2 trip 0   100MEG

*Change input logic signals into logic 0s or 1s
X7 trip B7 B7L Bitlogic
X6 trip B6 B6L Bitlogic
X5 trip B5 B5L Bitlogic
X4 trip B4 B4L Bitlogic
X3 trip B3 B3L Bitlogic
X2 trip B2 B2L Bitlogic
X1 trip B1 B1L Bitlogic
X0 trip B0 B0L Bitlogic

*Non-linear dependent source, B, for generating the DAC output
Bout Vout 0 V=((v(vrefp)-v(vrefm))/256)*(v(B7L)*128+v(B6L)*64+
+v(B5L)*32+v(B4L)*16+v(B3L)*8+v(B2L)*4+v(B1L)*2+v(B0L))+v(vrefm)
.ends

.subckt Bitlogic trip BX BXL
Vone    one     0       DC      1
SH      one     BXL     BX      trip    Switmod
SL      0       BXL     trip    BX      Switmod
.model switmod SW
.ends
*** END DAC Subcircuit **************************************

*** START ADC Subcircuit ************************************
.subckt ADC8bit VDD VREFP VREFM Vin B7 B6 B5 B4 B3 B2 B1 B0 CLOCK
* Set up common mode voltage
BCM     VCM     0       V=(V(VREFP)-V(VREFM))/2
* Set up logic switching point
R3      VDD     VTRIP   100MEG
R4      VTRIP   0       100MEG
* Ideal input sample and hold
XSH     VDD     VTRIP   VIN     OUTSH   CLOCK   SAMPHOLD
* Level shift by VREFM and 1/2LSB
BPIP    PIPIN   0       V=V(OUTSH)-V(VREFM)+((V(VREFP)-V(VREFM))/2^9)
* 8-bit pipeline ADC
X7      VDD     VTRIP   VCM     PIPIN   B7      VOUT7   ADCBIT
X6      VDD     VTRIP   VCM     VOUT7   B6      VOUT6   ADCBIT
X5      VDD     VTRIP   VCM     VOUT6   B5      VOUT5   ADCBIT
X4      VDD     VTRIP   VCM     VOUT5   B4      VOUT4   ADCBIT
X3      VDD     VTRIP   VCM     VOUT4   B3      VOUT3   ADCBIT
X2      VDD     VTRIP   VCM     VOUT3   B2      VOUT2   ADCBIT
X1      VDD     VTRIP   VCM     VOUT2   B1      VOUT1   ADCBIT
X0      VDD     VTRIP   VCM     VOUT1   B0      VOUT0   ADCBIT
.ends
* Ideal Sample and Hold subcircuit
.SUBCKT         SAMPHOLD        VDD     VTRIP   Vin     Vout    CLOCK
Ein     Vinbuf  0       Vin     Vinbuf  100MEG
S1      Vinbuf  VinS    VTRIP   CLOCK   switmod
Cs1     VinS    0       1e-10
S2      VinS    Vout1   CLOCK   VTRIP   switmod
Cout1   Vout1   0       1e-16
Eout    Vout    0       Vout1   0       1
.model switmod SW
.ends
* Pipeline stage
.SUBCKT ADCBIT  VDD     VTRIP   VCM     VIN     BITOUT  VOUT
S1      VDD     BITOUT  VIN     VCM     switmod
```

```
S2          0          BITOUT  VCM     VIN          switmod
Eouth       Vinh       0       VIN       VCM        2
Eoutl       Vinl       0       VIN       0          2
S3          Vinh       VOUT    BITOUT  VTRIP        switmod
S4          Vinl       VOUT    VTRIP     BITOUT     switmod
.model switmod SW
.ends
*** END ADC Subcircuit ************************************
.end
```

WinSpice results for the total noise.



Quantization noise = $V_{Qe,RMS}$ = 1.26mV

SNR measured is

$$SNR_{meas} = 20\log\frac{V_{p,meas}/\sqrt{2}}{V_{Qe,RMS}} = 20\log\frac{0.726/\sqrt{2}}{1.26mV} = 52.2dB$$

Which is very close to the 50dB ideal SNR calculated above. This result is approximately equal to the results of Example 31.2 were a 100MHz sampling frequency was used. This is what is expected. Changing the sampling frequency has no effect on the total quantization noise, and therefore no effect on the SNR. The deviation in the results is explained by the change in stepsize, necessary to get accurate results in this problem when using a 200MHz sampling frequency.

EE515: CMOS Mixed-Signal IC Design

Problem 31.4
Jim Slupe

31.4 Why is the amplitude of the tone at 45 MHz in the DAC output Spectrum shown in Fig. 31.3b smaller than the amplitude of the ADC input signal? What is the origin of the noise added to the DAC output signal in Fig 31.3b?

Answer:

The signal at 45 MHz is reduced by the sample and hold response of the ADC. Specifically, the 45 MHz input is at 0.75 V (-2.5 dB). The ADC sample and hold Sinc response will be as follows:

$$\text{Sinc}\,(\pi * f / fs\,) = \text{Sin}\,(\pi * f / fs\,) / (\pi * f / fs\,)$$

Taking the log of that result and multiplying by 20 gives you the difference between the input amplitude and the resulting amplitude of the signal:

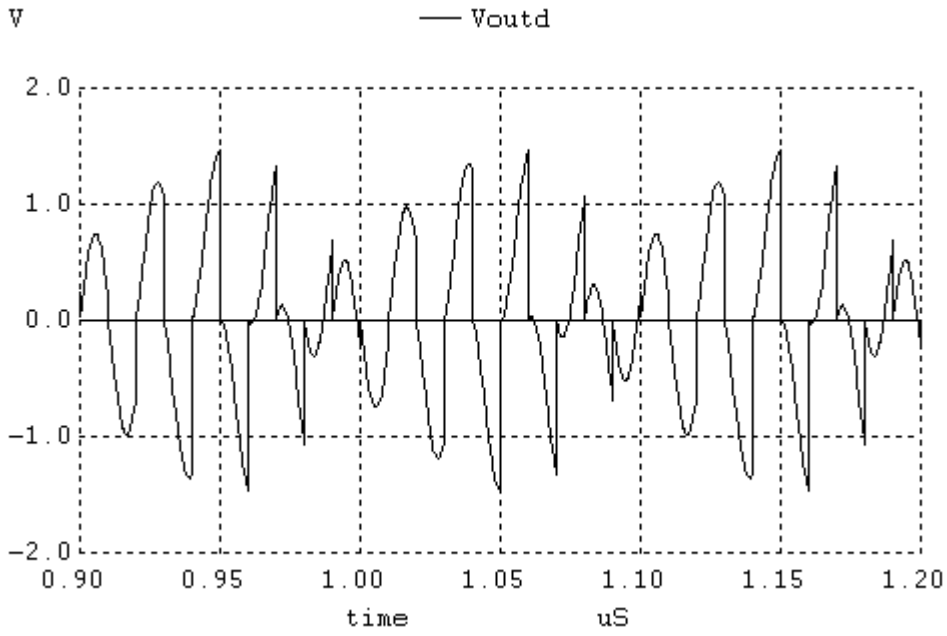
Vout(dB) = 20*log(Sin (pie*45 / 100) / (pie*45 / 100)) = -3.11 dB

Adding –3.11 dB to the input signal level of –2.5 dB gives a voltage level for the 45 MHz signal as –5.6 dB. This is in close agreement with the –5.5 dB value given in Example 31.3.

The noise added to the DAC output signal is quantization noise. What is interesting about the noise is the "binning" at 5 MHz intervals. This is a function of the way the sampling frequency and the input frequency interact. The following two lines were added to the SPICE file to highlight this interaction in the time domain:

```
*#plot Voutd xlimit 900n 1200n ylimit -1500m 1500m
Ediff Voutd 0 Vout Vin 1
```

Which resulted in the plot on the next page. The plot very clearly shows the signal at 5 MHz as a modulation envelope. The other frequencies can be deduced from the shape of the resulting sine wave.


V
Voutd
2.0
1.0
0.0
−1.0
−2.0
0.90
0.95
1.00
1.05
1.10
1.15
1.20
time
uS

David Goldman

EE 515 – CMOS Mixed-Signal IC Design

Problem 31.5

Question: When using Eq. (31.8) what is the assumed ADC input signal? Put your answer in terms of the ADC reference voltages.

Answer: The magnitude of the least significant bit is defined with the number of bits and the reference voltages as stated in (31.2)

$$V_{LSB} = 1LSB = \frac{V_{REF+} - V_{REF-}}{2^N}$$

With this definition, Eq. 31.8 assumes that input signal satisfies Bennett's criteria:

1. The signal must have a peak-to-peak voltage greater than 1 LSB.
2. The signal must be contained between V_{REF+} and V_{REF-}.
3. The signal must be "busy." In other words, input signal must periodically change the digital output.

EE515: CMOS Mixed-Signal IC Design
Brian Bergeson
Problem 31.6
bbergeso@poci.amis.com

Problem 31.6:

Describe in your own words the difference between specifying SNR and SNDR.

Answer:

ADC SNR (signal to noise ratio) is the ratio of the ADC input signal to the quantization noise introduced by the ADC (VQE,rms). SNR is calculated in dB with the following equation.

SNR = 20 * log (Vrms / VQE,rms)

ADC SNDR (signal to noise plus distortion ratio) is the ratio of the ADC input signal to the sum of the ADC quantization noise and the distortion introduced from nonlinear or non-ideal components in the ADC circuitry (VD,rms). SNDR is calculated, in dB, with the following equation.

SNDR = 20 * log (Vrms / (VQE,rms + VD,rms))

SNDR is the more practical calculation because real world components are non-ideal and create distortion.

31.7 Suppose a perfectly stable clock is available, the peak-to-peak clock jitter is zero in Eq. (31.12). Would we still have a finite aperture window if the clock has a finite risetime? Describe why or why not.

During the transitioning time of the clock, there is still uncertainty as to when the clock is considered high or low, so it is uncertain as to when the input signal is truly sampled. The slower the transitioning time of the clock, the larger the sampling uncertainty. Even though the peak-to-peak clock jitter is zero, we would still have a finite aperature window, if the clock has a finite risetime.

EE515: CMOS Mixed-Signal IC Design
Question 31.8
Richard Friel
Rich_Friel@AMIS.COM

Question: #31.08 How do the number of bits lost because of aperture jitter change with the frequency of the ADC input sinewave? If the ADC input is a DC signal, is aperture jitter a concern? Why?

Clock jitter is the variation in the period of the clock signal around the ideal value[1]. If we assume the input frequency is running at the Nyquist frequency, f_n, $(= f_s/2)$, then the sampling point is seeing the fastest transition in the input signal as shown in Figure 31.6 in the text. The slew rate of the signal, at the sampling point, is given by Equation (31.11) as follows:

$$\frac{\partial}{\partial t}(V_p \sin \pi f_s t) = \pi f_s V_p \underbrace{\cos \pi f_s t}_{=1} = \pi f_s V_p \tag{31.11}$$

We are interested in peak-to-peak jitter, defined as ΔT_s. Therefore, at time t=0, the cosine function has the maximum slope, as defined in Figure 31.6. Uncertainty in the sampling instant, ΔT_s, can be related to uncertainty in the sampled voltage, ΔV_s,

$$\frac{\Delta V_s}{\Delta T_s} = \pi f_s V_p \text{ or } \Delta V_s = \Delta T_s \pi f_s V_p \tag{31.12}$$

If the uncertainty is required to be at most, 0.5LSB, then maximum peak to peak clock jitter is determined to be,

$$\Delta T_s \leq \frac{1}{2^N} \frac{1}{\pi f_s} \tag{31.13}$$

[1] *The following solution is taken from the text, section (31.1.2), Clock Jitter.*

In the case where the signal being sampled has a frequency below the Nyquist value, Equation (31.13) cannot be used directly. We can rewrite Equation (31.13), using f_{in} ,

$$\Delta T_s \le \frac{1}{2^N}\frac{1}{2\pi f_{in}} \tag{31.15}$$

The SNR of the data converter is degraded from the ideal value, Eq. (31.14), when the input sampling clock is not ideal. Assuming a resolution loss >= 0.5 LSB, the following equation is derived from remembering that:

$$0.5\ \text{LSB} = (V_{ref+} - V_{ref-})/2^{N+1}, \text{ and}$$

$$Vp = (V_{ref+} - V_{ref-})/2, \text{ Then,}$$

$$\Delta T_s = \frac{1}{2^{N-N_{LOSS}}}\frac{1}{2\pi f_{in}} \tag{31.17}$$

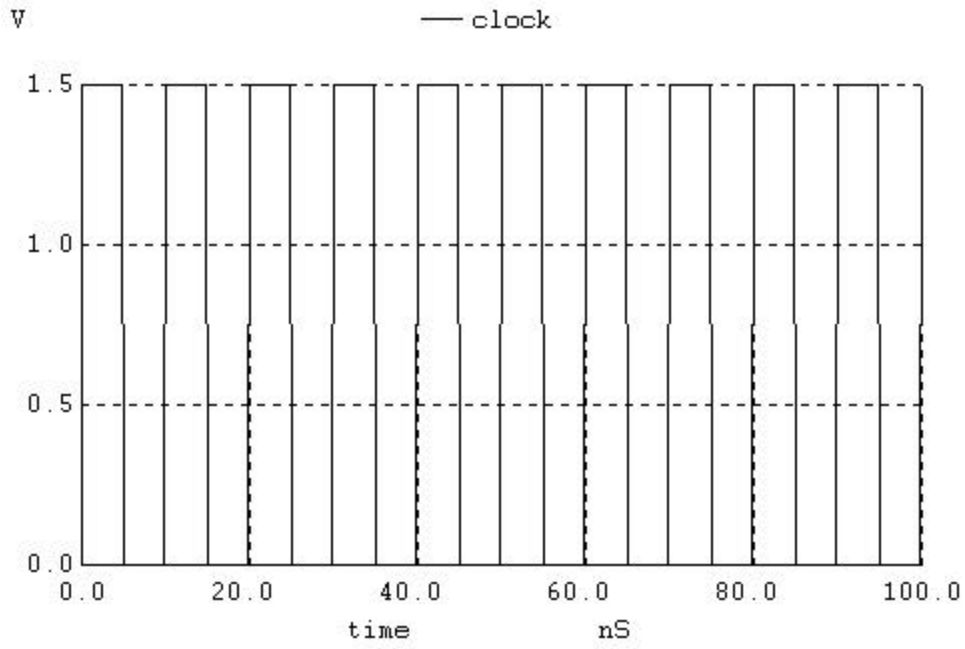
By solving for N_{Loss} , term in the above equation,

$$N_{LOSS} = (\ln(\Delta T_S 2\pi f_{in})\frac{1}{\ln(2)}) + N \tag{31.17.1}$$

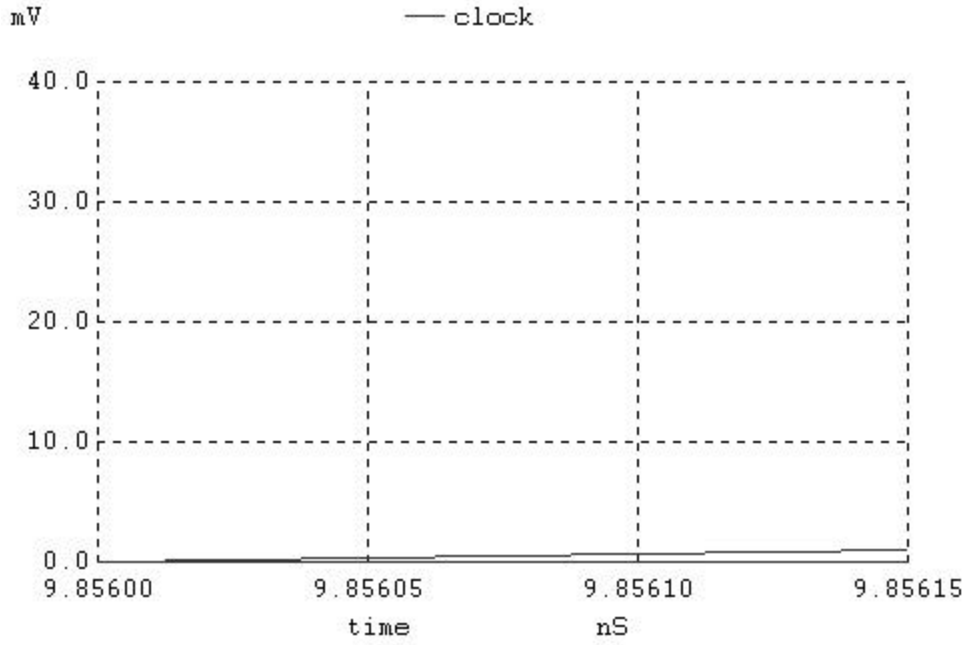
we can observe how the number of bits lost, N_{LOSS}, changes with frequency of the input, for a known aperture jitter, ΔT_s . If the ADC input is a DC signal, the aperture jitter, ΔT_s =0, because all of the samples are at the same point and f_{in} = 0. Therefore, aperture jitter is not a concern at DC or f_{in} = 0.

Problem 31.9 Show the time-domain signal that generates the spectrum shown in Fig. 31.10. Verify in the time-domain that the signal's rising and falling edges do indeed vary from their ideal positions.

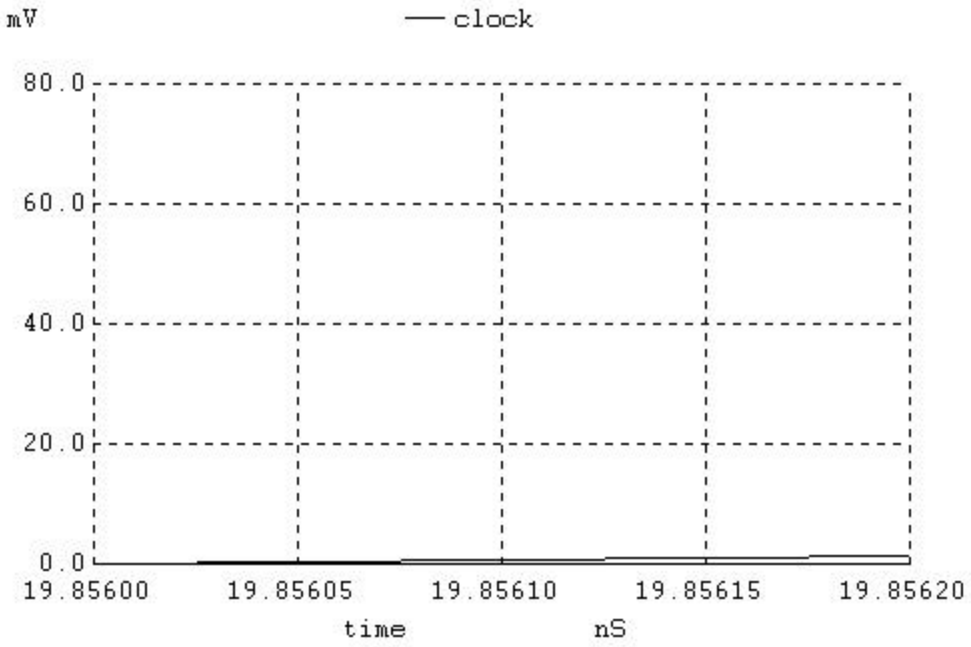
Solution:
The clock generated from Fig. 31.10 is given below, followed by three consecutive rising and falling edges.



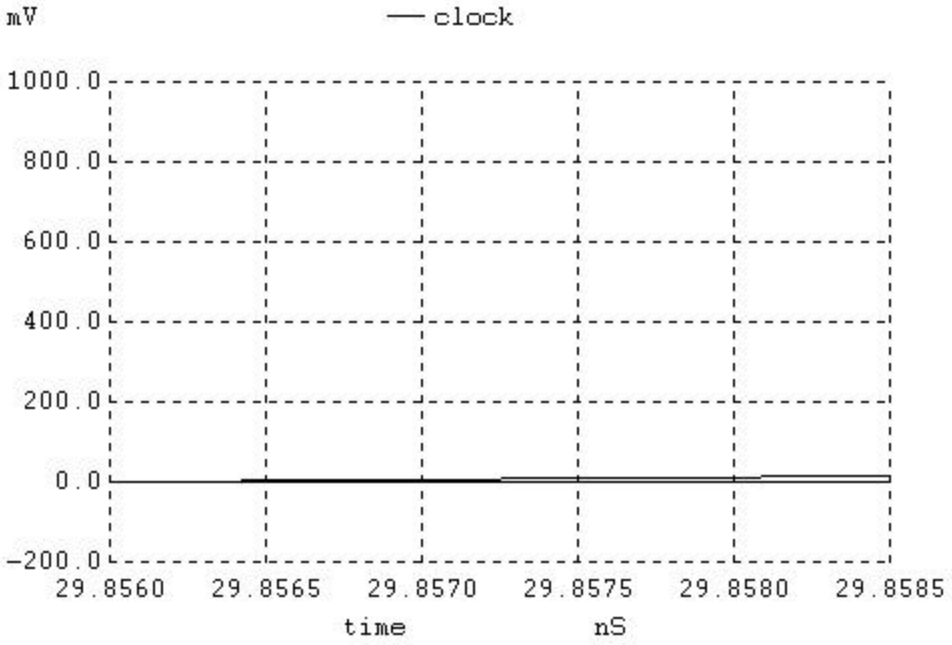
Three consecutive rising edges:



9.856015 nS

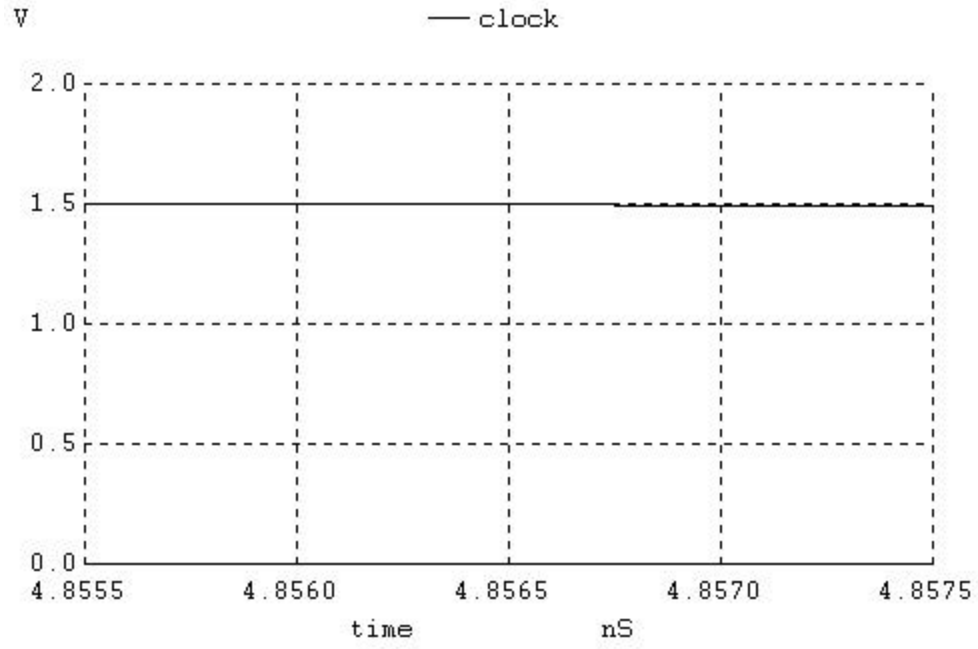


19.856025 nS

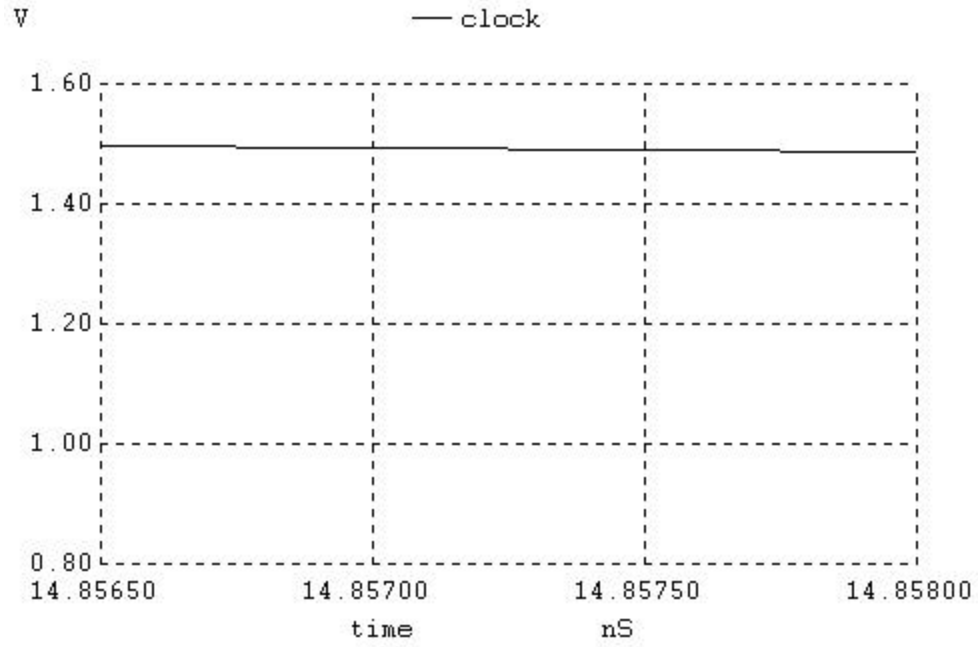


29.8564 nS

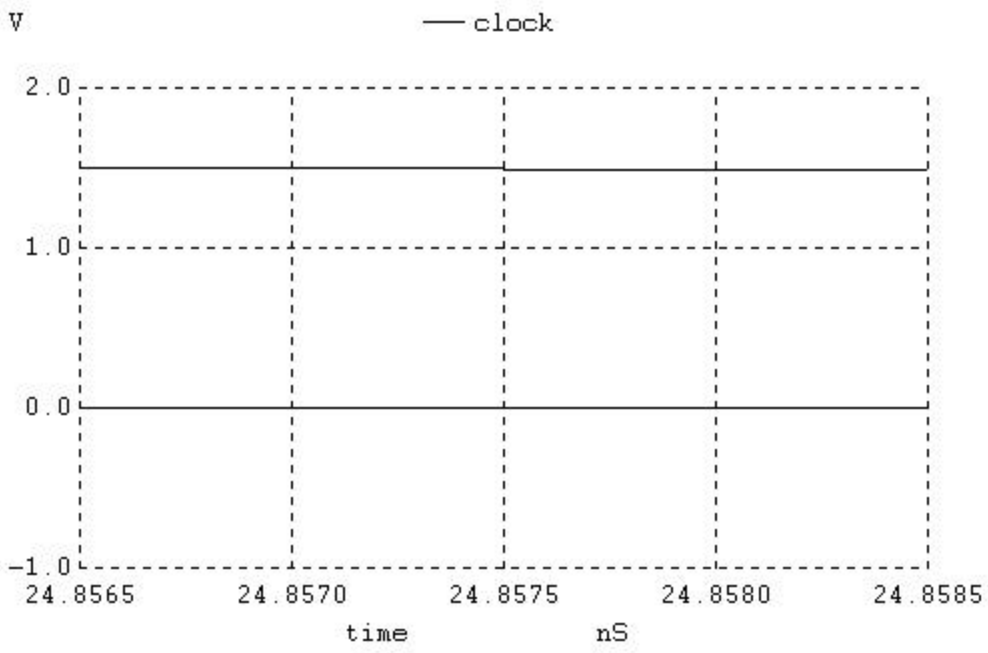
Three consecutive falling edges:



4.8567 nS



14.85659 ns



24.8575 nS

For an ideal period(no jitter), the rise/fall displacement from the period boundary would be uniform. The table below shows that the rise and fall times vary from these ideal positios.

$ÄT = T_{boundary} - T_{rising/falling}$

The rise boundary is some multiple of 10 nS, while the fall boundary is 5 nS later/earlier.

T_{rising}	$ÄT_{rising}$	$T_{falling}$	$ÄT_{falling}$
9.856015 nS	ÄT=143.99 pS	4.8567 nS	ÄT=143.3 pS
19.856025 nS	ÄT=143.98 pS	14.85659 nS	ÄT=143.41 pS
29.8564 nS	ÄT=143.6 pS	24.8575 nS	ÄT=142.5 pS

This table shows a variations of the rising edge for three consecutive edges, and a shift back and forth of the falling edge. QED

Bob Moehrke EE515 (bmoehrke@amis.com)

Question 31.10

Describe in your own words the problems with simulating clock jitter using SPICE.

While clock jitter is truly random; since SPICE uses a computer, the closest it can come to this is pseudo-random. Following equation 31.28 (which requires that FS, the rate at which the clock jitter varies, be 1/simulation time) will lessen the appearance of the jitter being periodic, but it is still pseudo-random at best. Another difficulty with using SPICE is that if the jitter to be modeled is on the order of picoseconds the step size of the simulation will have to be very small, thus making simulation times excessively long.

Jeremy Rice

31.11

The Autocorrelation function gives us the average value of a signal as a function of time.

Find the ACF for a 1V DC signal

$$R_{in}(t) = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} V_{in}(\tau) \cdot V_{in}(\tau + t)\, d\tau$$

$V_{in} = 1$

$$R_{in}(t) = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} 1 \cdot 1\, d\tau$$

$$= \lim_{T_0 \to \infty} \frac{1}{T_0} \left[\tau\right]_{-T_0/2}^{T_0/2}$$

$$= \lim_{T_0 \to \infty} \frac{1}{T_0} \left[T_0/2 - (-T_0/2)\right]$$

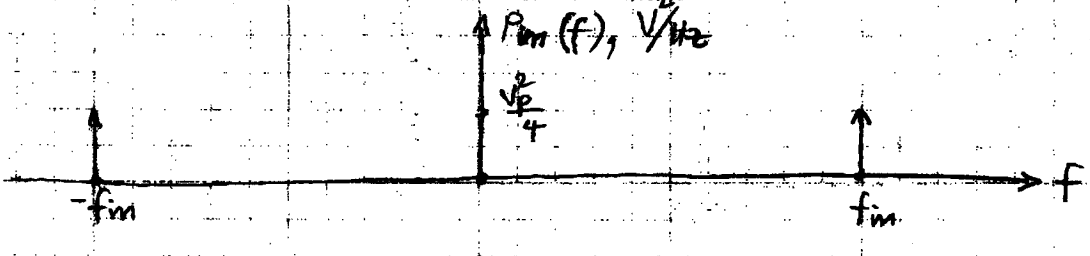
$$R_{in}(t) = \lim_{T_0 \to \infty} \frac{1}{T_0} \cdot T_0 = 1$$

Problem 31.12 Plot the power spectral density of a sinewave. From this plot show how to determine the average and RMS values of the sinewave. Show the procedure for both one-sided and two-sided spectrums.

using equations (31.29) $V_{in}(t) = V_p \sin 2\pi f_{in} t$

The power spectral density is equation (31.36)

$$P_{in}(f) = \frac{V_p^2}{4} \cdot [\delta(f+f_{in}) + \delta(f-f_{in})]$$



Total average power is equation (31.37)

$P_{avg} = \int_{-\infty}^{\infty} P_{in}(f) \cdot df = 2 \cdot \int_0^{\infty} P_{in}(f) \cdot df$ assuming 1Ω load.

Since the areas of delta functions are 1 at ± f_{in}, P_{avg} can be obtained directly from the above plot just by doubling its magnitude

$P_{avg} = 2 \cdot V_p^2/4 = V_p^2/2$ W for one-sided spectrum

$$P_{avg} = \int_{-\infty}^{\infty} P_{in}(f)\, df = \int_{-\infty}^{0} P_{in}(f)\, df + \int_0^{\infty} P_{in}(f)\, df$$

$$P_{avg} = \frac{V_p^2}{4} + \frac{V_p^2}{4} = \frac{V_p^2}{2}$$ for two-sided spectrum

$$V_{RMS} = \sqrt{P_{avg}} = \sqrt{2\int_0^{\infty} P_{in}(f) \cdot df}$$ equation (31.38)

$$V_{RMS} = \sqrt{\frac{V_p^2}{2}} = \frac{V_p}{\sqrt{2}}\ V$$ for one-sided spectrum

$$= \sqrt{\frac{V_p^2}{4} + \frac{V_p^2}{4}} = \sqrt{\frac{V_p^2}{2}} = \frac{V_p}{\sqrt{2}}$$ for two sided spectrum

Tyler J. Gomm
tjgomm@micron.com

31.13 Sometimes the average power specified by Eq. (31.37) is termed *total average normalized* power of a signal. Why?

Equation (31.37) states that the average signal power, *assuming a 1-W load*, is given by

$$P_{AVG} = \int_{-\infty}^{\infty} P_{in}(f) \cdot df = 2 \cdot \int_{0}^{\infty} P_{in}(f) \cdot df \qquad \text{(units, } V^2/\Omega \text{ or } Watts\text{)}$$

Note that this equation is a sum (integration) of power across frequency, and is a total of average power at various frequencies.

The average voltage value of a sinewave is zero. Therefore the average value cannot be used to calculate the power of a sinewave. This is the reason that RMS values are used when considering power. The RMS value of a sinewave will dissipate the same amount of power as a DC voltage of the same value when either is applied across a resistance.

$$\frac{V_{RMS}^{\ 2}}{R} = \frac{V_{DC}^{\ 2}}{R} \quad \text{when} \quad V_{RMS} = V_{DC}$$

This is to say that the RMS value squared is the average power of the sinewave (assuming a 1-Ω load). When the average power (RMS squared) of each sinewave in a spectrum is summed up across the spectrum (or in other words, we integrate the power spectral density), this is called the total average power.

Equation (31.37) may also be referred to as *total average normalized* power, because the assumption of a 1-Ω load *normalizes* the power to the load. For example, normalized DC power would be calculated as follows

$$P_{NORMALIZED} = \frac{V^2 / R}{1 / R} = V^2$$

This is the same as assuming a 1-Ω load.

By normalizing with respect to the load, power may be compared from system to system without considering the load.

David Goldman

EE 515 – CMOS Mixed-Signal IC Design

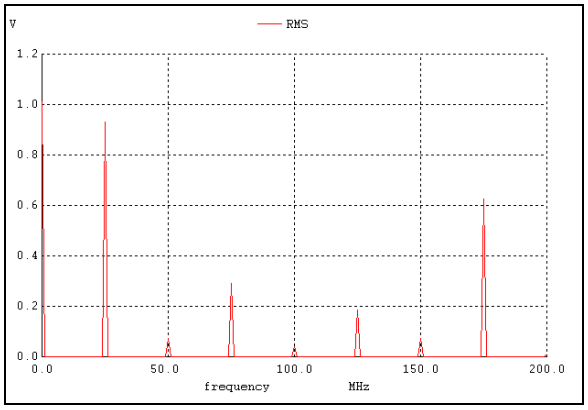
Problem 31.14

Question: When WinSPICE generates a plot from an FFT the units on the y-axis are volts peak (the peak value of a sinewave at a given frequency). How do we change this plot into RMS voltages, voltage spectral density, and power spectral density vs. frequency?

Answer: The following examples are taken from WinSPICE example file 31.2. Only the WinSPICE command scripts are shown.

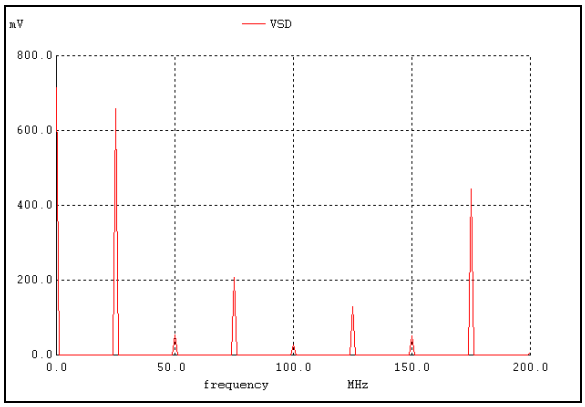
In order to plot the RMS voltage, the peak values must be divided by the square root of two. This is shown below.

```
*#destroy all
*#run
*#spec 0 200MEG 1MEG Vout
*#let m = mag(Vout)
*#let RMS = m/0.707
*#plot RMS
```



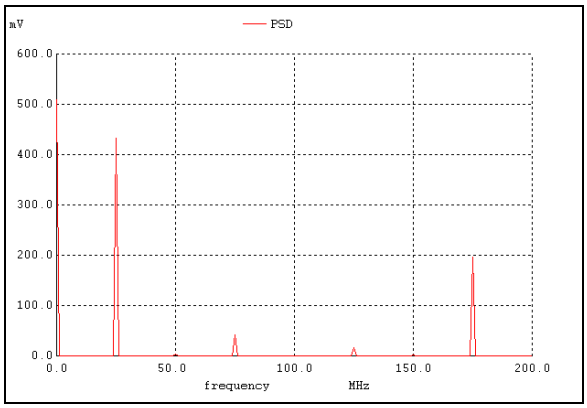
The voltage spectral density has units of $V/(Hz)^{1/2}$. Since the FFT uses discrete values, a bandwidth of 1 Hz can be assumed for each individual sinewave. Voltage spectral density is then the same as peak volts. This is shown below.

```
*#destroy all
*#run
*#spec 0 200MEG 1MEG Vout
*#let m = mag(Vout)
*#let VSD = m
*#plot VSD
```



Finally, power spectral density (PSD) is the voltage spectral density squared (employing the previous assumption.) The command script and plot are shown below.

```
*#destroy all
*#run
*#spec 0 200MEG 1MEG Vout
*#let m = mag(Vout)
*#let VSD = m
*#let PSD = VSD*VSD
*#plot PSD
```



Gexin

31.15 Repeat Ex. 31.12 if the sinewaves are first sampled.

Example 31.12
Determine the ACF, PSD, Average Power, and RMS value of a signal, V(t) made up of three sine waves with peak amplitudes, of V_1, V_2, and V_3 with frequencies of f_1, f_2 and f_3.

Solution for 31.15:

Using equation (31.41) the ACF is

$$R(t)=\frac{V_1^{\,2}}{2}\cdot\cos(2\pi f_1\cdot nT_s)+\frac{V_2^{\,2}}{2}\cdot\cos(2\pi f_2\cdot nT_s)+\frac{V_3^{\,2}}{2}\cdot\cos(2\pi f_3\cdot nT_s)\qquad\text{(units, } V^2\text{)}$$

The PSD is determined using equation (31.42)

$$P(f)=\frac{V_1^{\,2}}{4T_s}\sum_{k=-\infty}^{\infty}[\delta(f-f_1+kf_s)+\delta(f+f_1+kf_s)]+\frac{V_2^{\,2}}{4T_s}\sum_{k=-\infty}^{\infty}[\delta(f-f_2+kf_s)+\delta(f+f_2+kf_s)]$$
$$+\frac{V_3^{\,2}}{4T_s}\sum_{k=-\infty}^{\infty}[\delta(f-f_3+kf_s)+\delta(f+f_3+kf_s)]$$

Assume the signal is passed through an ideal reconstruction filter (RCF) with a bandwidth of $f_s/2$.

The PSD of the signal, after passing through the RCF, has amplitude of $V_1^{\,2}/4$ at frequencies of $\pm f_1$, has amplitude of $V_2^{\,2}/4$ at frequencies of $\pm f_2$ and has amplitude of $V_3^{\,2}/4$ at frequencies of $\pm f_3$.

The average power of this signal is given by:

$$P_{avg}=\frac{V_1^{\,2}+V_2^{\,2}+V_3^{\,2}}{2}\qquad\text{(units, Watts)}$$

Finally, the RMS value for this signal is given by:

$$V_{RMS}=\sqrt{P_{avg}}=\sqrt{\frac{V_1^{\,2}+V_2^{\,2}+V_3^{\,2}}{2}}\qquad\text{(units, V)}$$

Brandon Roth
brandonroth@micron.com

31.16 Suppose the jitter in a clock signal can be characterized using the PDF shown in Fig. 31.12. Further if T_s = 100ps estimate the RMS value of clock jitter, standard deviation, and variance of the jitter.

Variance is defined as the average of the square of a signals departure from its average value. If we define the average value of the PDF in Fig. 31.12 as

$$\bar{y} = \int_{-\Delta T_s/2}^{\Delta T_s/2} t \frac{1}{\Delta T_s} dt = 0$$

In general the variance, ó^2 is then defined by

$$\sigma^2 = \overline{(y-\bar{y})^2} = \int_{-\infty}^{\infty} (y-\bar{y})^2 \cdot \rho(y) \cdot dy$$

Substituting for the PDF in Figure 31.12

$$t = y$$
$$\bar{y} = 0$$
$$\rho(y) = \frac{1}{\Delta T_s}$$

$$\sigma^2 = \int_{-\Delta T_S/2}^{\Delta T_S/2} t^2 \cdot \frac{1}{\Delta T_S} \cdot dt = \frac{1}{3\Delta T_s}\left[t^3\right]_{-\Delta T_s/2}^{\Delta T_s/2} = \frac{1}{3\Delta T_s}\left[\frac{2\Delta T_s^{\,3}}{8}\right] = \frac{\Delta T_s^{\,2}}{12}$$

RMS is defined as standard deviation, and is the square root of the variance. Therefore the standard deviation or RMS clock jitter is

$$\sigma = \frac{\Delta T_s}{\sqrt{12}}$$

If T_s = 100ps then the variance is

$$\sigma^2 = \frac{(100\,ps)^2}{12} = 8.33\times10^{-22}\ (\text{seconds}^2).$$

And the RMS jitter or standard deviation is

$$\sigma = \frac{100\,ps}{\sqrt{12}} = 2.887\times10^{-11}\ (\text{seconds}).$$

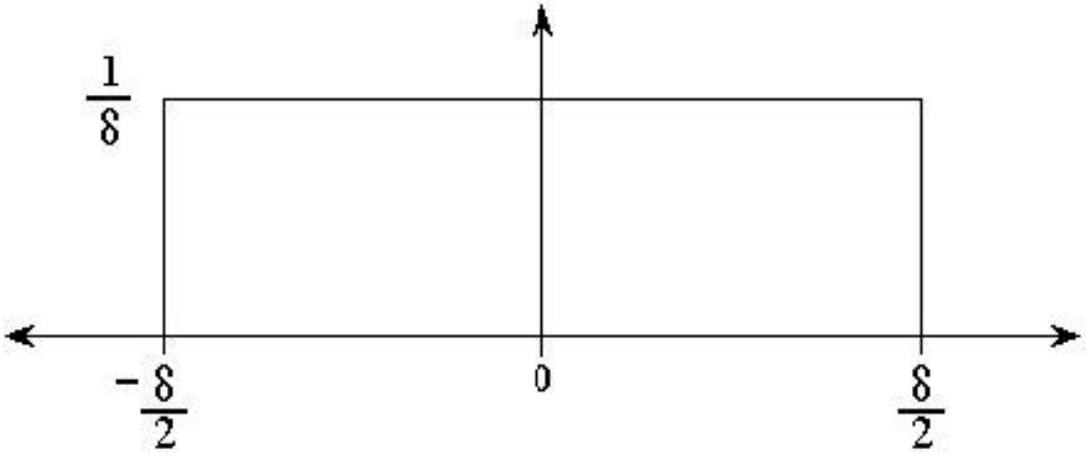
EE515: CMOS Mixed-Signal IC Design

Problem 31.17
Jim Slupe

31.17 Suppose that a noise voltage has the PDF shown in Fig. 31.12. If the maximum voltage deviation from the ideal value is 10 uV estimate the RMS value of the noise (the standard deviation) and the noise power (the variance).

Answer:

Looking first at the variance (since the standard deviation will simply be the square root of the variance), the maximum deviation from the ideal value (0 uV) I will call $\delta/2$ (equal to 10 uV). Figure 31.12 can be redrawn for this problem as follows:



$$\sigma^2 = \int_{-d/2}^{d/2} V^2 \bullet 1/\delta \bullet dV = \frac{1}{3 \bullet \delta}\left(V^3\right)\Big|_{-d/2}^{d/2} = \frac{\delta^2}{12} = 33.33...\text{pW (noise power)}$$

$$\sigma = \delta/\sqrt{12} = 5.77 \text{ uV (noise voltage)}$$

This probably looks familiar and it is. The area under the PDF is equal to 1 or $1/\delta \bullet (\delta/2 - (-\delta/2)) = 1$ This was the case with Figure 30.47 as well as Figure 31.12 and graphically depicts an equal probability of an occurrence without any bias relative to the 0 value.

Gexin

#31.18 Suppose that a noise voltage has a Gaussian PDF as seen in Fig. 31.13. If the maximum voltage deviation from the ideal value is 10uV, estimate the RMS value of the noise (the standard deviation) and the noise power (the variance).

Solution for 31.18:

The RMS value of the noise is $\boldsymbol{s} = \frac{peak\_to\_peak\_value}{6} = \frac{2 \times 10uV}{6} = 3.33uV$

The noise power of the noise is $\boldsymbol{s}^2 = (3.33uV)^2 = 11.11 \times 10^{-12} V^2$

EE515: CMOS Mixed-Signal IC Design
Brian Bergeson
Problem 31.19
bbergeso@poci.amis.com

Problem 31.19:
Repeat Ex. 31.1 if we want to include an error from sampling jitter, Pave_jitter, of 1uW.

Example 31.1:
Determine the effective number of bits for an ADC with Vref+ = 1.5V, Vref- = 0V, and a measured (quantization error/noise) Vqe_rms of 2mV.

If we assume that the input peak amplitude, Vp, is 0.5 * (Vref+ - Vref-) or 0.75V,
then the measured SNR is given by:
SNR = (0.75 / sqrt(2)) / 2mV = 265 = 48.5dB

The effective number of bits, Neff, is (from Eq. 31.5) 7.76 bits.

Solution:
For this problem the SNR must be recalculated using the total rms noise. The total rms noise is the rms quantization noise (Vqe_rms) and the rms noise from the clock jitter (Vjit_rms).

If we assume:
Pave_jitter = Vjit_rms^2 / R, where R = 1ohm

then we know that:
Vjit_rms = sqrt(Pave_jitter) = sqrt(1uW * 1ohm) = 1mV

Next we must calculate the total rms noise:
Vtot_rms = sqrt(Vqe_rms^2 + Vjit_rms^2) = sqrt(2mV^2 + 1mV^2) = 2.24mV

Now we can recalculate SNR using Vtot_rms, instead of Vqe_rms:
SNR = (0.75 / sqrt(2)) / Vtot_rms = 237, which is 47.5dB.

The new Neff (using Eq 31.5) is then 7.6 bits.

31.20 If a DC signal is input to a data conversion system, is Eq. (31.51) valid? Name three conditions on the input signal in order for this equation to be valid.

If a DC signal is input to a data conversion system, Eq. (31.51) is not valid, since the input signal is not busy. In order for Eq. (31.51) to be valid, the following conditions (Bennett's criteria) must be met:

1) The input signal's amplitude can not exceed the limits of the ADC's power supplies. Exceeding these limits adds spikes to the output spectrum of the ADC, affecting the quantization noise spectrum.

2) The input signal amplitude must be much greater than the ADC's LSB. If this is not the case, the output of the ADC, after converting back to an analog signal, can appear square-wave-like. This adds spurs or spikes to the output spectrum.

3) The input signal must be busy. In other words, two consecutive digital output codes can not be the same.

EE515: CMOS Mixed-Signal IC Design
Question 31.21
Richard Friel
Rich_Friel@AMIS.COM

Question: #31.21

If the standard deviation of the quantization noise in a data conversion system equals 1mV, using Eq. (31.56), plot the PSD of the quantization noise.

Comment on the assumption that the noise power is limited to the Nyquist frequency, f_n. Does this result in an over- or underestimate for the actual power in the spectrum of interest?

Solution:

From Eq. (31.54) we know that if the quantization noise is considered an evenly distributed random variable, the standard deviation can be found by:

$$\sigma = \frac{V_{LSB}}{\sqrt{12}} = \sqrt{2\int_0^{\infty} P_{Qe}(f)df} \qquad (31.54)$$

The standard deviation is the square root of the variance. In order to find the variance, the thus use Eq. (31.56), we square the standard deviation. In this case, we find that the variance is equal to 1mV and is the standard deviation squared from Eq. (31.56),

$$\sigma^2 = \frac{V^2{}_{LSB}}{12} = 2\int_0^{\frac{f_s}{2}} P_{Qe}(f)\cdot df \qquad (31.56)$$

If the quantization noise is considered a random variable and is evenly distributed, the quantization noise power is constant and is equal to,

$$P_{Qe}(f) = \frac{1}{f_s}\frac{V^2_{LSB}}{12} \tag{31.57}$$

$\boxed{\sigma^2 = \frac{V^2{}_{LSB}}{12} = (1mV)^2}$, and the QNPSD is plotted in Figure 1.

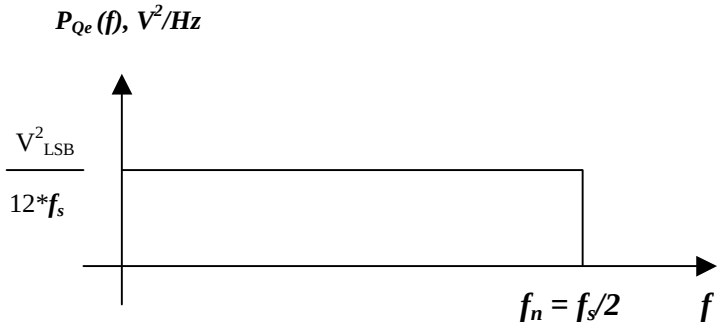


Figure 1. Quantization Noise Power Spectral Density

We know that the quantization noise does not experience aliasing because quantization occurs after sampling. So while it is correct to look at a wide spectrum to calculate noise, it is more useful to limit our view of the spectrum to the frequencies up to the Nyquist frequency, ***(= f_n = f_s/2),*** where the desired signal spectrum should reside. We do this by assuming that the entire noise power lies in the base spectrum that is described by Eq. (31.56).[1]

The assumption that the entire quantization noise resides in the base spectrum results in an over-estimation of the actual noise power in the spectrum of interest.

[1] Fig 1. And Text From the course notes, CMOS Mixed Signal IC Design by Jake Baker, pp. 88.

Problem 31.22: Show why averaging two 8-bit words, as seen in Fig. 31.19, must result in a 9-bit word. (Why isn't the sum of the two words divided by two [the average] another 8-bit word?)

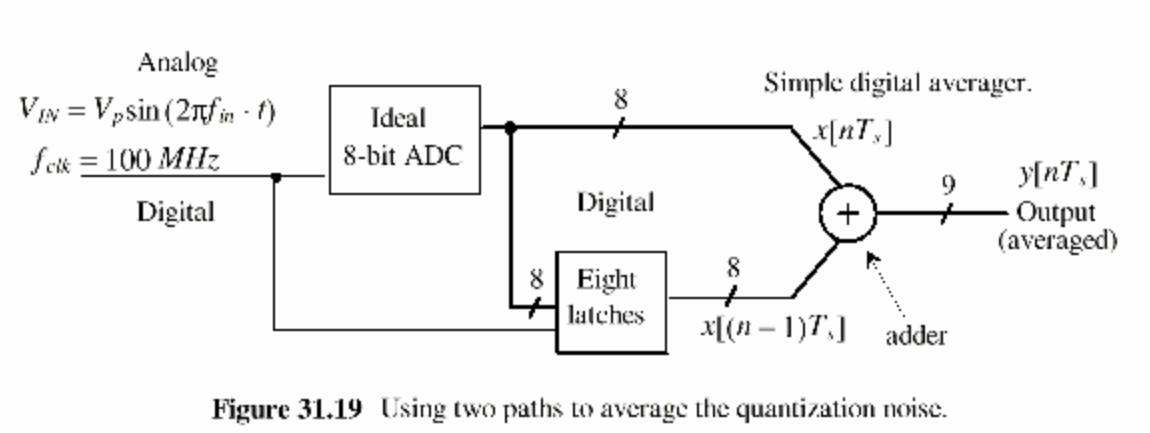
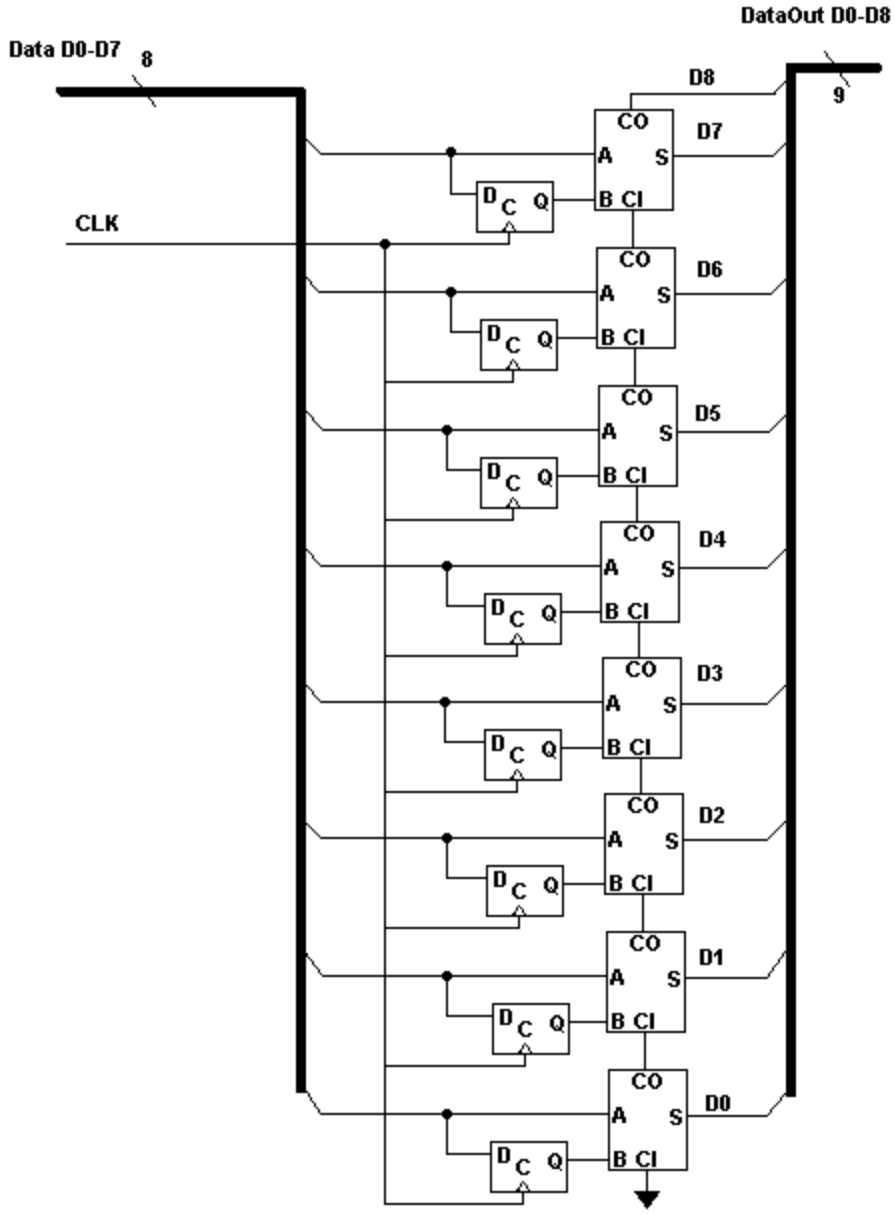


Figure 31.19 Using two paths to average the quantization noise.

Solution: As in the figure below, two 8-bit words are added together. The carry out from the addition operation becomes the MSB in a 9-bit word. The actual arithmetic average could be created by ignoring D0 (the LSB) and just using the upper 8 bits. However, the extra bit could be retained, keeping the extra information, keeping in mind that K = 2 (averaging two samples).

```
  1010 0101 = 165
  1010 0000 = 160
  ------------  ----
1 0100 0101 = 325
```

Shifting the result to the right, 325 becomes 162 (dropping the LSB from 162.5 to 162) keeping only 8 bits. Keeping the extra bit gives a more accurate picture of the true average. (Indeed, 1010 0010 .1 is 162.5, the true average.)


DataOut D0-D8
Data D0-D7
8
9
D8
D7
D6
D5
D4
D3
D2
D1
D0
CLK
CO
A
S
B CI
D
C
Q

QED

Bob Moehrke EE515 (bmoehrke@amis.com)

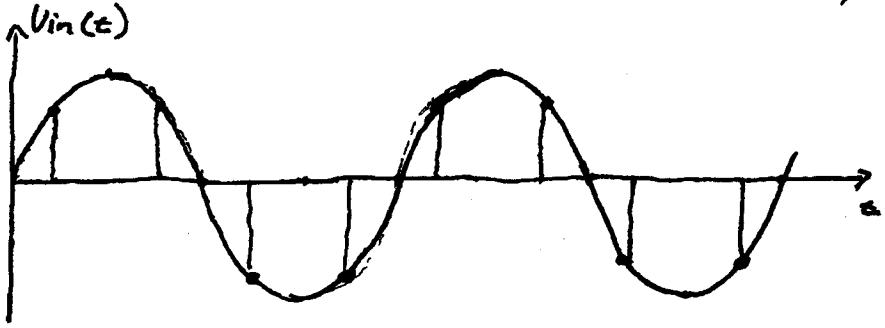
Question 31.23

Why must Bennett's criteria be valid for the averaging to reduce the quantization noise in Fig. 31.19? Give an example where averaging will not reduce quantization noise.

If Bennett's criteria do not hold (the input signal isn't moving around much), it is likely that a sample will be the same as the previous one. If this happens frequently then all of the benefits of averaging (K=2) will disappear. The idea is to find a midpoint (or average, K=2) between 2 samples in order to find a new quantized point that can be different than the A/D by itself. If 2 consecutive points are the same, the average will be that point itself. No new information is gained. The simplest example of this is a DC input signal.

Jeremy Rice

31.24



$V_{in}(n)$ with $f_{in} = f_s/4$

Using the filter given by 31.59:

$$H(z) = \frac{Y(z)}{X(z)} = 1 + z^{-1}$$

The magnitude is given by Eq (31.61)

$$|H(z)| = \sqrt{\left(1 + \cos\left[2\pi \frac{f_{in}}{f_s}\right]\right)^2 + \left(\sin\left[2\pi \frac{f_{in}}{f_s}\right]\right)^2} \Bigg|_{f_{in}/f_s = 1/4}$$

$$= \sqrt{(1+0)^2 + 1^2}$$

$$|H(z)|_{\frac{f_{in}}{f_s} = 1/4} = \sqrt{2}$$

Problem 31.25

Using eq.(31.68) $V_{Qe,RMS} = \frac{1}{\sqrt{K}} \cdot \frac{V_{LSB}}{\sqrt{12}} = \frac{1}{\sqrt{K}} \cdot \frac{2V_p/2^N}{\sqrt{12}}$

with eq.(31.1) $SNR_{ideal} = 20\log_{10} \frac{V_p/\sqrt{2}}{\frac{1}{\sqrt{K}} \cdot \frac{2V_p/2^N}{\sqrt{12}}}$

$$= 20\log_{10}\left(\sqrt{K} \cdot \sqrt{12} \cdot \frac{2^N}{2\sqrt{2}}\right)$$

$$= 20\log_{10} 2^N + 20\log_{10}\frac{\sqrt{3}}{\sqrt{2}} + 20\log_{10} K^{\frac{1}{2}}$$

$$SNR_{ideal} = 6.02N + 1.76 + 10\log_{10} K \quad (dB)$$

For a slow or DC input signal, averaging is not beneficial. The input signal must be changing by at least 1 LSB between sampling instances in order to benefit from averaging. Ideally, we want an output from averaging to fall in the middle of two adjacent codes. For example, $V_{out} = \frac{V_A + V_B}{2}$; if $V_A = V_B$, then $V_{out} = V_A = V_B$; averaging here has no benefit. For K=1, the SNR equation above would reduce to equation (31.4), which is non-averaged. Eq. (31.4) is derived based on $V_{Qe,RMS} = \frac{V_{LSB}}{\sqrt{12}}$, and it is true if Bennett's criteria holds. Otherwise, $V_{Qe,RMS}$ would be different from $V_{LSB}/\sqrt{12}$. So eq. (31.4) and the SNR derived above are not valid for the slow or DC input signal.

Tyler J. Gomm
tjgomm@micron.com

31.26 Assuming Bennett's criteria are valid, does averaging ADC outputs (or DAC inputs) put any restrictions on the bandwidth of the input signal? Why? Give an example.

When ADC outputs (or DAC inputs) are averaged, there are several benefits, including:

- Reduced quantization error
- Reduced effects resulting from jitter in the sample clock (sampling error amplitude power, $P_{AVG,\,jitter}$)
- Relaxed requirements for the Anti-Aliasing Filter (AAF)
- Increased SNR

However, in order for these advantages to be realized, the bandwidth of the input signal must be restricted.

The effective RMS quantization noise voltage (when averaging is used) is given in Eq. (31.68), where K is the number of terms averaged:

$$V_{Qe,RMS} = \frac{1}{\sqrt{K}} \cdot \frac{V_{LSB}}{\sqrt{12}} \tag{31.68}$$

Eq. (31.68) assumes the averaging filter does *not* attenuate the input signal. If the bandwidth is *not* limited, the SNR can be *worse* than the SNR of the system *without* averaging. For example, if the input frequency of the system shown in Fig. 31.19 were $f_s/3$, then the RMS amplitude of the desired signal would be $V_p/\sqrt{2}$ instead of the ideal $2 \cdot V_p/\sqrt{2}$ (the averaging gain of 2 is lost, because the filter gain is 1 at $f_s/3$). This means the desired signal amplitude is *reduced* by a factor of 2, and the SNR of the system is made *worse* than if averaging had not been used at all!

Averaging results in attenuation of many of the input signal frequencies, and an average will go to zero when the input frequency is $f_s/2$. In particular, to realize an improvement in the SNR and the quantization error, equation (31.69) must be used to limit the input bandwidth, B:

$$B = \frac{f_s \diagup 2}{K} = \frac{f_n}{K} \quad and \quad f_{in} \le B \tag{31.69}$$

David Goldman

EE 515 – CMOS Mixed-Signal IC Design

Problem 31.27

Question: Comment on the statement "The factor of two in the magnitude response of Fig. 31.22 at low-frequencies simply indicates that the digital word length increases by 1-bit."

Answer: A digital word length of 8-bits contains 256 unique numbers. By increasing the digital word length to 9-bits, the number of possible unique numbers that can be represented increases to 512. In fact, any digital word that increases by one bit can now represent twice as much information. Figure 31.22 demonstrates this by showing the peak magnitude as twice the magnitude of the word.

Gexin

31.28: What is the magnitude response of $Z^{-2}+Z^{-3}$?

Solution for 31.28:

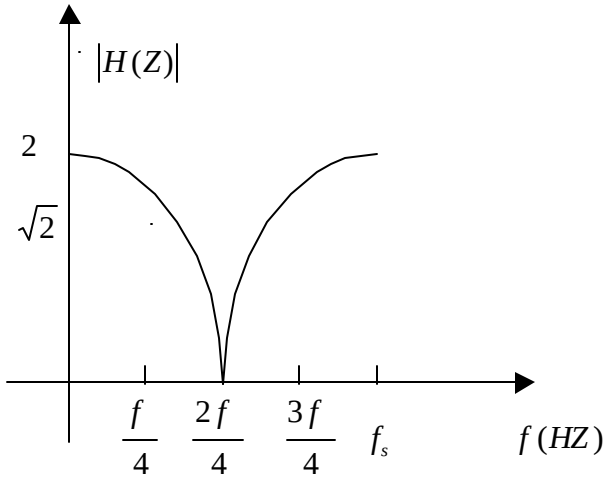
Let $H(Z)=Z^{-2}+Z^{-3}=\dfrac{1+Z}{Z^3}$

This transfer function has 3 poles at origin and one zero at Z= -1.

The magnitude of this transfer is:

$$|H(Z)|=\frac{\text{Product\_of\_distances\_to\_zeros}}{\text{Product\_of\_distances\_to\_poles}}$$

At DC, distance to zero is 2, the distance to poles is 1. We get $|H(Z)|=2$ at DC.
At $f_s/4$ or $3f_s/4$, distance to zero is $\sqrt{2}$, the distance to poles is 1. So $|H(Z)|=\sqrt{2}$ at $f_s/4$ or $3f_s/4$.
At $f_s/2$, distance to zero is 0, the distance to poles is 1. We get $|H(Z)|=0$ at $f_s/2$.
The magnitude response of the transfer function is shown below:



Brandon Roth
brandonroth@micron.com

31.29 Repeat Example 31.15 if 16 ADC outputs are averaged, that is K=16.

From Example 31.15, determine the SNR of the data converter (8-bit ADC clocked at 100MHz) with 100ps of peak-to-peak jitter in the input sampling clock. Assuming the ADC's input is a full-scale sinewave at 25MHz, and the clock jitter has a Gaussian PDF.

The SNR is

$$SNR = 20\log\frac{V_p/\sqrt{2}}{RMS\quad Noise}$$

Where the RMS noise is the sum of the quantization noise and the RMS clock jitter.

First the quantization noise for averaged ADC outputs, from equation 31.51 is

$$V_{Qe,RMS} = \frac{1}{\sqrt{K}}\cdot\frac{V_{LSB}}{\sqrt{12}} = \frac{V_{REF+}-V_{REF-}}{2^N\sqrt{12K}} = \frac{1.5}{2^8\sqrt{12(16)}} = 0.423mV\ .$$

Then the clock jitter for a Gaussian PDF is found using equation 31.47

$$P_{AVG,jitter} = \left(\frac{\Delta T_s}{6}\right)^2\cdot\frac{\left(V_p\cdot 2\pi f_{in}\right)^2}{2} = \left(\frac{100ps}{6}\right)^2\cdot\frac{\left(0.75\cdot 2\pi\cdot 25MHz\right)^2}{2} = 1.93\times10^{-6}V^2.$$

Then the RMS voltage associated with clock jitter is

$$jitter_{RMS} = \sqrt{P_{AVG,jitter}} = 1.39mV\ .$$

The RMS noise is then the sum of the RMS jitter and the quantization noise. To add two RMS values, the values must first be converted to power, summed, and then converted back to an RMS value by taking the square root.

$$RMSNoise = \sqrt{jitter^2{}_{RMS} + V^2{}_{Qe,RMS}} = \sqrt{1.39^2 + 0.423^2}\,mV = 1.45mV.$$

Finally

$$SNR = 20\log\frac{0.75/\sqrt{2}}{1.45mV} = 51.26dB$$

Notice by sampling 16 ADC outputs the quantization noise is reduced by about a fourth, the clock jitter remains the same, and the SNR is increased.

EE515: CMOS Mixed-Signal IC Design

Problem 31.30
Jim Slupe

31.30 How accurate does an 8-bit ADC have to be in order to use a digital filter to average 16 output samples for a final output resolution of 10-bits (see Eq. [31.53]). Assume the ideal LSB of the 8-bit converter is 10 mV. Your answer should be given in both mV and % of the full-scale.

Answer:

If K = 16 then the increased resolution from sampling is 10 log K divided by 6.02 or 12.04 / 6.02 or 2 using equation 31.53. This in turn yields an effective 10 bit final resolution. If an LSB is 10 mV then the full range of the ADC is: $(2^8 - 1)$ * 10 mV or 2.55 volts. Now if this same voltage range is divided by $2^{10} - 1$ the size of an LSB is 2.49 mV or 0.098 % of full scale.

This value can be put back into equation 31.1 as a way of testing the result. The following result is obtained:

$$SNR_{IDEAL} = 20 * \log \frac{1.275V/\sqrt{2}}{2.49mV/\sqrt{12}} = 61.96dB$$

or the ideal SNR for a 10-bit ADC. This is something of a circular argument, but it does prove out the consistency of the equations. Unfortunately it doesn't end there.

As stated in the text: "The ADC output should, ideally, change in increments of the exact LSB voltage. In reality, the changes will be different from the ideal output levels (as just discussed.) In order to achieve an increase in the number of final bits, the output of the ADC must be accurate (its actual levels must be spaced from the ideal levels) to within

$$\pm \frac{Vref_+ - Vref_-}{2^{Nfinal+1}} = \pm(0.5LSB) * \frac{1}{2^{Ninc}} \qquad (31.75)$$

where no averaging (Ninc = 0 and K = 1) means the ADC is at least 0.5 LSBs accurate." Therefore the actual accuracy requirement of the ADC is:

$$\pm(0.5 * 10mV) * \frac{1}{2^2} = \pm 1.25mV \text{ or } \pm 0.049\% \text{ of full scale}$$

Brandon Roth
brandonroth@micron.com

31.31 If a DC signal is applied to a data converter can a digital averaging filter be used to increase the system's resolution? What about if a dither signal is added to the DC input? Use simple time-domain drawings to illustrate your answers.

Figure 1 and 2 show the outputs of the ADC and digital averaging filter. When a DC signal is applied to the ADC the digital code with the closest voltage representation is the resulting output. The averaging filter constantly averages the same Nth digital code from the output of the ADC. As seen in figure two no increase in resolution is realized, where the Nth digital code of the ADC is equal to the Xth digital code of the averaging filter. Then digital averaging filter can not increase the system's resolution if a DC signal is applied to the input.

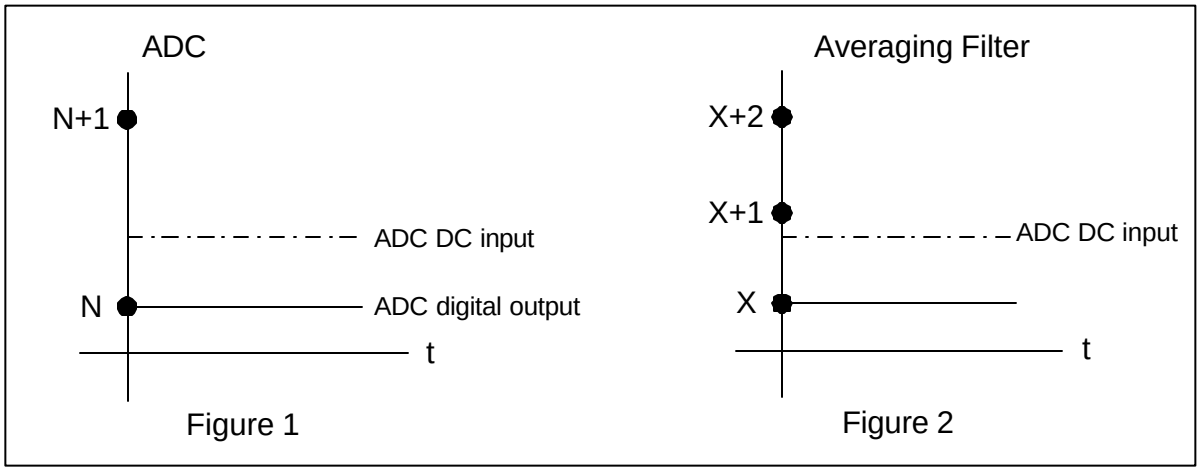


Figure 1 Figure 2

In figure 3 dither is added to the ADC DC input signal. The amplitude of the dither is at least one LSB, and it is centered about the DC input signal. Figure 4 shows that by averaging the dithered DC signal a closer digital representation of the original signal is realized. The X+1 digital code is a more accurate representation of the DC input signal than the Xth digital code in figure 2. Therefore, adding dither to a DC input signal allows the digital averaging filter to increase the resolution of the system.

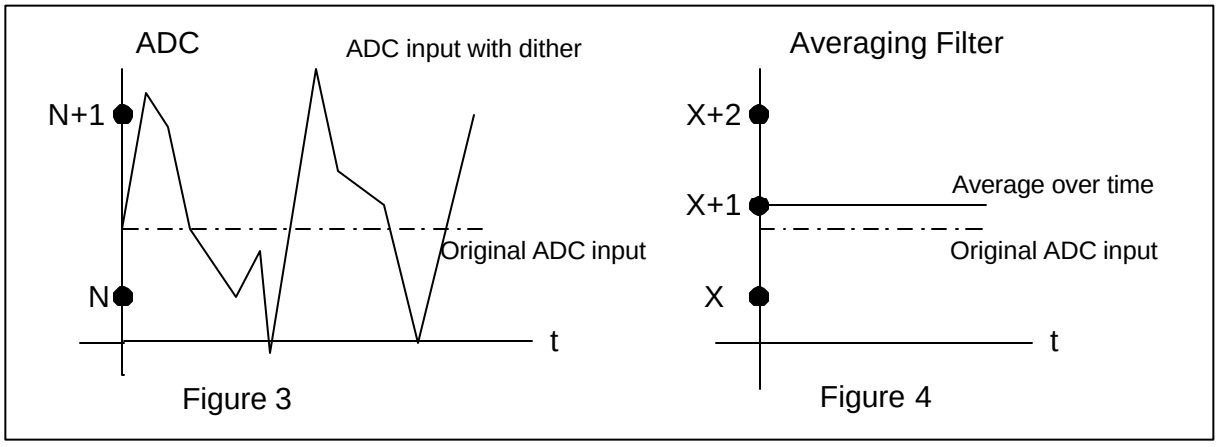


Figure 3 Figure 4

EE515: CMOS Mixed-Signal IC Design
Brian Bergeson
Problem 31.32
bbergeso@poci.amis.com

Problem 31.32:
Name three characteristics of all digital filters.

1) All digital filters are periodic with sampling frequency.

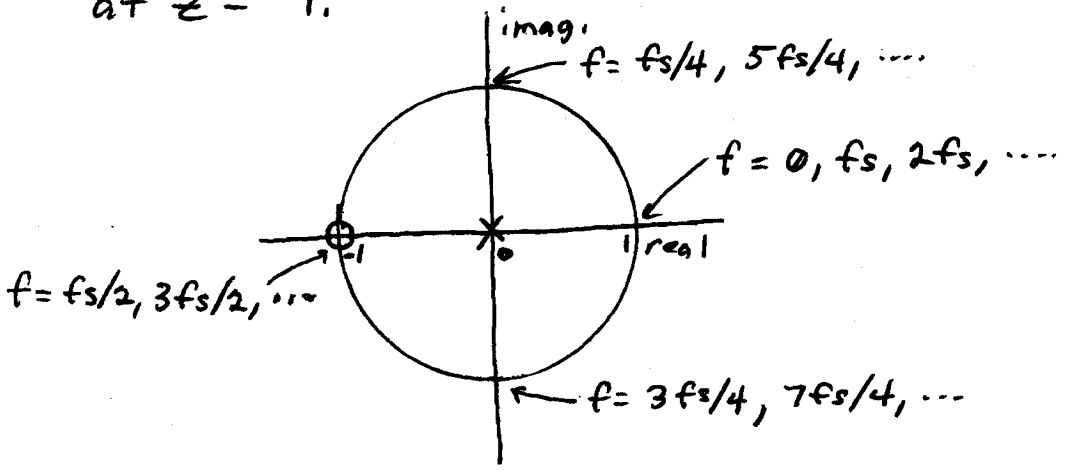
2) In order for a digital filter to be practical (can be implemented) the number of poles in its transfer function must be greater than the number of zeros in its transfer function.

3) A pole or zero at the origin of the Z-plane, in a digital filter's transfer function, does not affect the magnitude response of the digital filter.

31.33 Plot equation (31.59) on a z-plane. Using this plot, show how to graphically determine the magnitude and phase response shown in Fig. 31.22

$$H(z) = \frac{Y(z)}{X(z)} = 1 + z^{-1} \quad (31.59)$$

$H(z) = 1 + z^{-1} = \frac{z+1}{z}$, therefore there is a pole at $z = 0$ and a zero at $z = -1$.



$$|H(f)| = \frac{\text{distance to zero}}{\text{distance to pole}} \quad (31.79)$$

at: $f = 0$, $|H(f)| = \frac{2}{1} = 2$

$f = fs/4$, $|H(f)| = \frac{\sqrt{2}}{1} = \sqrt{2}$

$f = fs/2$, $|Hf)| = \frac{0}{1} = 0$

$f = 3fs/4$, $|H(f)| = \frac{\sqrt{2}}{1} = \sqrt{2}$

31.33 (cont.)

$\angle H(f) = \angle$ of zero $- \angle$ of pole (31.80)

at: $f = 0$, $H(f) = 0° - 0° = 0°$

$f = f_s/4$, $H(f) = 45° - 90° = -45°$

$f = f_s/2$, $H(f) = 90° - 180° = -90°$ (just above the zero)

$f = f_s/2$, $H(f) = -90° + 180° = 90°$ (just below the zero)

$f = 3f_s/4$, $H(f) = -45° + 90° = 45°$

The plots, using these magnitudes and angles, are shown in Fig. 31.22.

EE515: CMOS Mixed-Signal IC Design
Question 31.34
Richard Friel
Rich_Friel@AMIS.COM

Question: #31.34

The magnitude response shown in Fig. 31.34 becomes infinite as the input signal approached DC. Since the filter is digital, what is the maximum output of the filter?

Solution:

The maximum output of an ADC, using a two's complement binary offset format is equal to: (VDD – 1 LSB), and is represented in Fig 31.37 in the text as 0111 1111. This represents +127 in two's complement format or 1.494V for the particular ADC as described in Fig. 31.37.

If the ADC is connected to a digital filter, the digital filter will saturate, (i.e. reach maximum value), at DC to 1111 1111, assuming that filter contains the same number of bits as the ADC output and that there is a summing feedback loop in the digital filter.

Problem 31.35 Show that the peak (+127) and valley (-128) amplitudes of the two's complement signals in Fig. 31.37 sum to –1.

Solution: In two's complement, +127 (8 bits) is 0111 1111; -128 (also 8 bits) is 1000 0000. Therefore the sum (still 8 bits) is:

```
  0111 1111
  1000 0000
  ---------
  1111 1111
```

In two's complement, 1111 1111 is indeed –1. QED

Bob Moehrke EE515 (bmoehrke@amis.com)

Question 31.36

Summarize the method of changing a number from binary-offset to two's complement. Demonstrate addition and subtraction using two's complement numbers. Show how, in two's complement, 8, 33, and 111 sum to 152. Assume a 10-bit word size.

Going from binary-offset to two's complement is done by complimenting the MSB of the word. Using 4 bit numbers, adding 2 & 3 (=5) is done simply by normal binary addition:

```
  0010
+ 0011
  0101
```

Subtracting 2 from 3 (=1) is done by first complimenting the number to be subtracted. This is done by complimenting each digit and then adding one. This complimented number is then added to the other number to get the final result of the subtraction:

```
  0011
- 0010  ->  1101  ->  (+1) = 1110
```

and changing the "–" to a "+":

```
  0011
+ 1110
  0001
```

Using a 10-bit word size, 8 + 33 + 111 = 152:

```
  0000001000
  0000100001
+ 0001101111
  0010011000
```

Prob 31.37

Find the transfer function for a digital filter that sums 16 inputs and then outputs the total. Then, given a sampling frequency of 100MHz, plot the magnitude response of the filter.

$$\frac{Y(z)}{X(z)} := 1 + z^{-1} + z^{-2} .... + z^{-15}$$

$$H(z) := \frac{\left(1 - z^{-1}\right)}{\left(1 - z^{-1}\right)} \cdot \left(1 + z^{-1} + z^{-2} + ... + z^{-15}\right)$$

So, the final transfer function, H(z), is given by:

$$H(z) := \frac{1 - z^{-16}}{1 - z^{-1}}$$

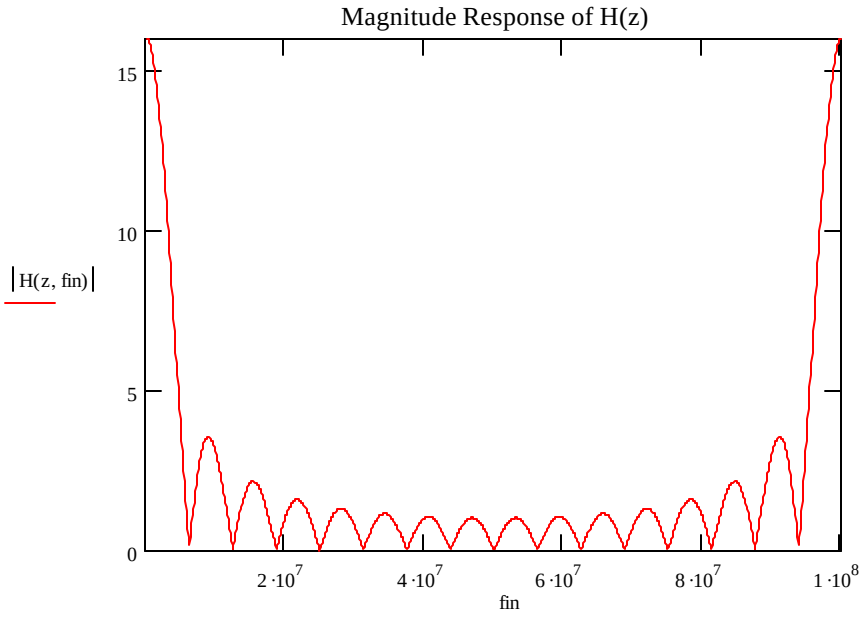
Substituting:

$$j := \sqrt{-1} \quad fs := 100 \cdot 10^{6}$$

$$z(fin) := e^{j \cdot 2 \cdot \pi \frac{fin}{fs}}$$

We now have H defined as a function of z which is in turn a function of fin, and fs

$$H(z, fin) := \frac{1 - z(fin)^{-16}}{1 - z(fin)^{-1}}$$



Problem 31.38 Comment on the benefits and drawbacks of using an averaging filter with and without decimation.

With Decimation:

Benefits:

Using averaging filter with decimation reduces the sampling frequency from f_s to f_s/K, where K is a number of input samples used for averaging to increase the resolution of data converter. The lower sampling rate causes power dissipation to be smaller than before decimation. Also, circuit designing can be simpler.

Drawbacks:

The maximum amount of attenuation or the ratio of main lobe to first side lobe approaches the limit of 13.5 dB as averaging factor K increases. To obtain an attenuation greater than 13.5 dB at frequencies above f_s/K requires cascading of averaging filter stages. Sampling frequency coming out of the final stage can be quite low, from f_s/K to f_s/K^L, where L is the number of stages ; this in turn causes a narrow input bandwidth that is not very useful. Also, drooping at the maximum input bandwidth is not reduced after cascading the required number of averaging filter stages.

Without Decimation:

Benefits:

The maximum attenuation increases without limiting the maximum input frequency bandwidth as in the case of cascading the averaging filters with decimation.

Drawbacks:

Greater droop exists at the maximum input bandwidth than with decimation.

Tyler J. Gomm
tjgomm@micron.com

31.39 Verify the z-domain function specified by Eq. (31.100) has a frequency response given by Eq. (31.101). How are the typical input and output signals in the time domain related for this filter?

Eq. (31.100) is the transfer function of a cascade of L averaging filters.

$$H(z) = \left[\frac{1}{K} \cdot \frac{1 - z^{-k}}{1 - z^{-1}} \right]^L \tag{31.100}$$

Substituting $z = e^{j \cdot 2p \frac{f}{f_s}}$

$$H(f) = \left[\frac{1}{K} \cdot \frac{1 - e^{-j \cdot K \cdot 2p \frac{f}{f_s}}}{1 - e^{-j \cdot 2p \frac{f}{f_s}}} \right]^L$$

We know that

$$\left|1 - e^{-j \cdot x}\right| = \left|(1 - \cos x) + j \sin x\right| = \sqrt{(1 - \cos x)^2 + (\sin x)^2}$$

$$\sqrt{(1 - \cos x)^2 + (\sin x)^2} = \sqrt{1 - 2\cos x + \cos^2 x + \sin^2 x} = \sqrt{2(1 - \cos x)}$$

(Using the trigonometric identity: $\sin^2 x + \cos^2 x = 1$).

Now $|H(f)|$ may be written as

$$|H(f)| = \left[\frac{1}{K} \cdot \frac{\sqrt{2\left(1 - \cos K \cdot 2p \frac{f}{f_s}\right)}}{\sqrt{2\left(1 - \cos 2p \frac{f}{f_s}\right)}} \right]^L$$

Using the trigonometric identity: $\sin^2 x = \frac{1}{2}(1 - \cos 2x)$, $|H(f)|$ becomes

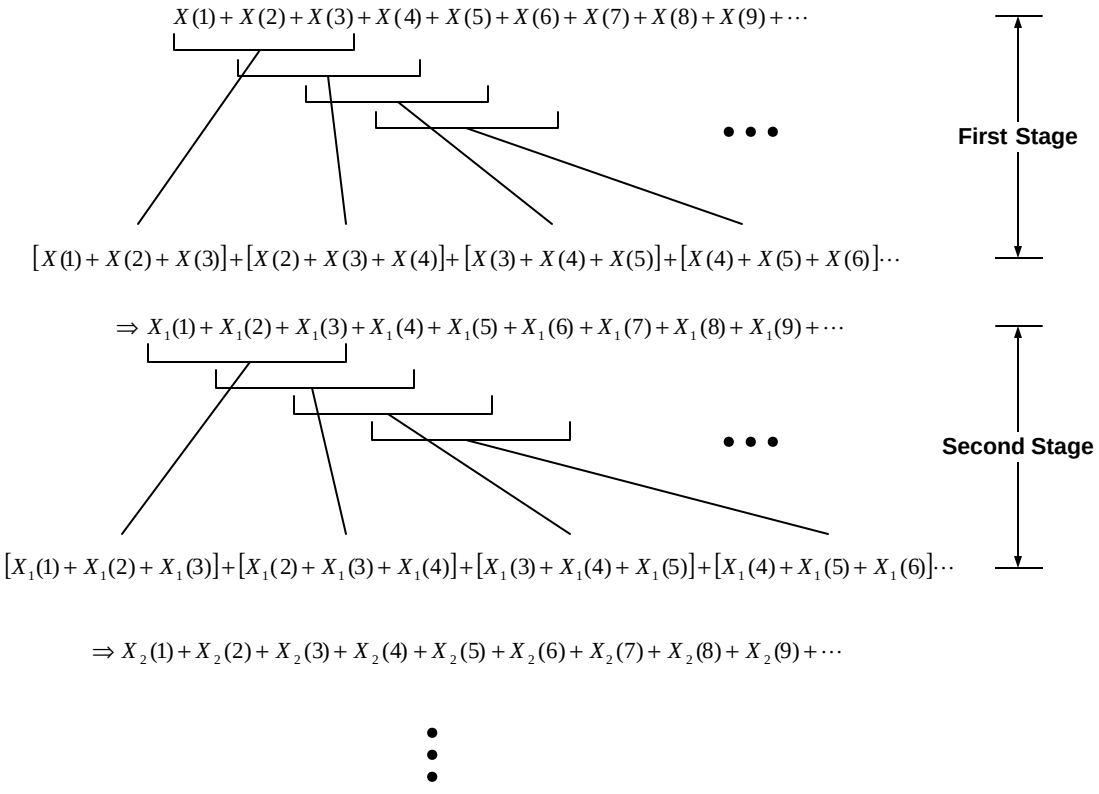
$$|H(f)| = \left[\frac{1}{K} \cdot \frac{\sqrt{4 \cdot \sin^2 Kp \frac{f}{f_s}}}{\sqrt{4 \cdot \sin^2 p \frac{f}{f_s}}} \right]^L = \left[\frac{1}{K} \cdot \frac{2 \cdot \sin Kp \frac{f}{f_s}}{2 \cdot \sin p \frac{f}{f_s}} \right]^L$$

The 2 in the numerator and denominator cancel. Now multiply the top and bottom by $p\,\frac{f}{f_s}$

$$|H(f)| = \left[\frac{p\,\frac{f}{f_s}}{Kp\,\frac{f}{f_s}} \cdot \frac{\sin Kp\,\frac{f}{f_s}}{\sin p\,\frac{f}{f_s}}\right]^L = \left[\frac{\mathrm{sinc}\left(\mathrm{K}p\,\frac{f}{f_s}\right)}{\mathrm{sinc}\left(p\,\frac{f}{f_s}\right)}\right]^L$$

Assuming this transfer function is implemented as a cascade of moving average filters (rather than a cascade of accumulate-and-dump circuits), the typical output is delayed by $L \cdot K \cdot T_s$ (constant delay, therefore linear phase), and the output frequency is equal to the input frequency (no decimation has occurred).

A sequence with K=3 may be visualized as follows. Only two stages are shown (L=2):



David Goldman

EE 515 – CMOS Mixed-Signal IC Design

Problem 31.40

Question: What is the magnitude response of $(1-z^{-1})^3$. Sketch a block diagram implementation for this filter.

Answer: A simple comb filter is $(1-z^{-1})$ and can be written as

$$H(z) = \left(1 - z^{-1}\right) = 1 - \frac{1}{z} = \frac{z}{z} - \frac{1}{z} = \frac{z-1}{z}$$

The following equation is an example of how to determine the zeros of H(z).

$$1 \cdot e^{j0} = 1$$

From this example, an angle of 0 produces a 1. This 1 indicates a zero in H(z). Therefore, the zeros will be located at each of the following: 0 (or DC), f_s, $2f_s$, $3f_s$, etc. A plot of the z-plane is shown in Figure 1.

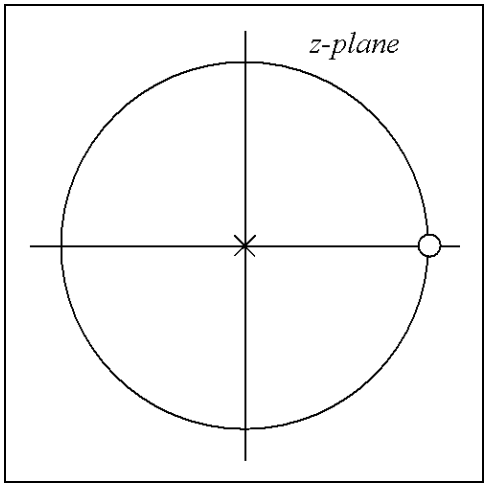


Figure 1: A Simple Comb Filter with H(z)=1-z^{-1}

Now, three simple comb filters cascaded together can be written as

$$\left(1 - z^{-1}\right)^3 = \left(\frac{z-1}{z}\right)^3 = \left(\frac{z-1}{z}\right) \cdot \left(\frac{z-1}{z}\right) \cdot \left(\frac{z-1}{z}\right)$$

The z-plane of this equation is shown is Figure 2.

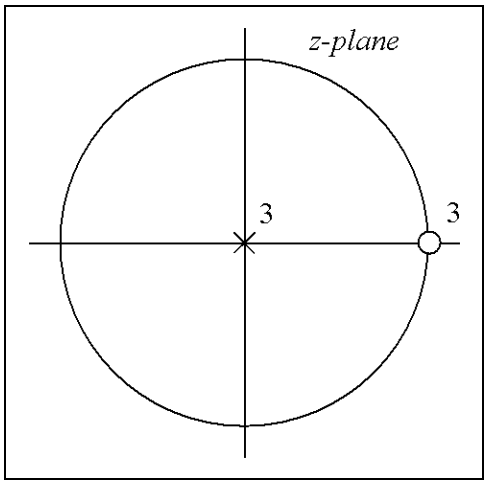


Figure 2: A Comb Filter with $H(z)=(1-z^{-1})^3$

This filter has the effect of averaging the digital signal. The general frequency response of an averaging filter is given as 2^L. In this example, L = 3, therefore $2^L = 8$. An example of this is shown below in Figure 3.

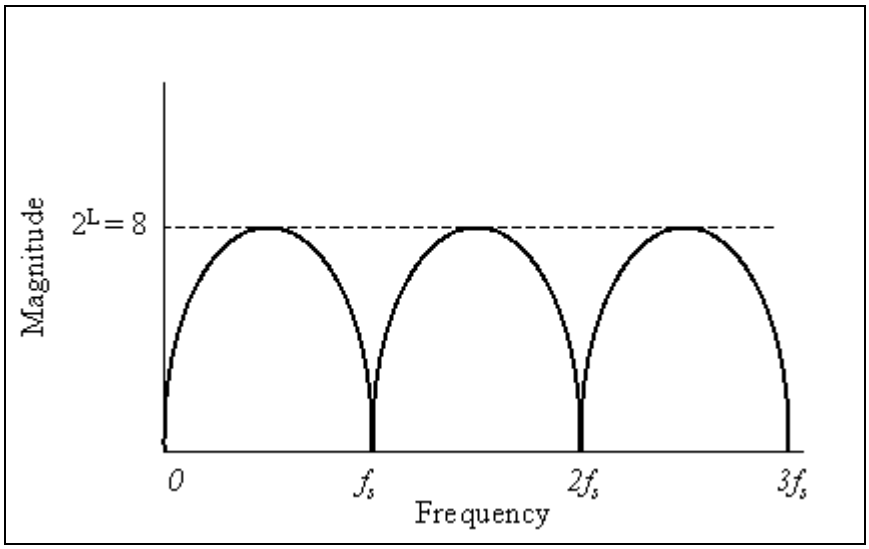


Figure 3: Magnitude Response of Cascaded Comb Filter

A block diagram of a single comb filter is shown in Figure 31.50 of the text. A cascade of three comb filters cascaded is shown below in Figure 4.

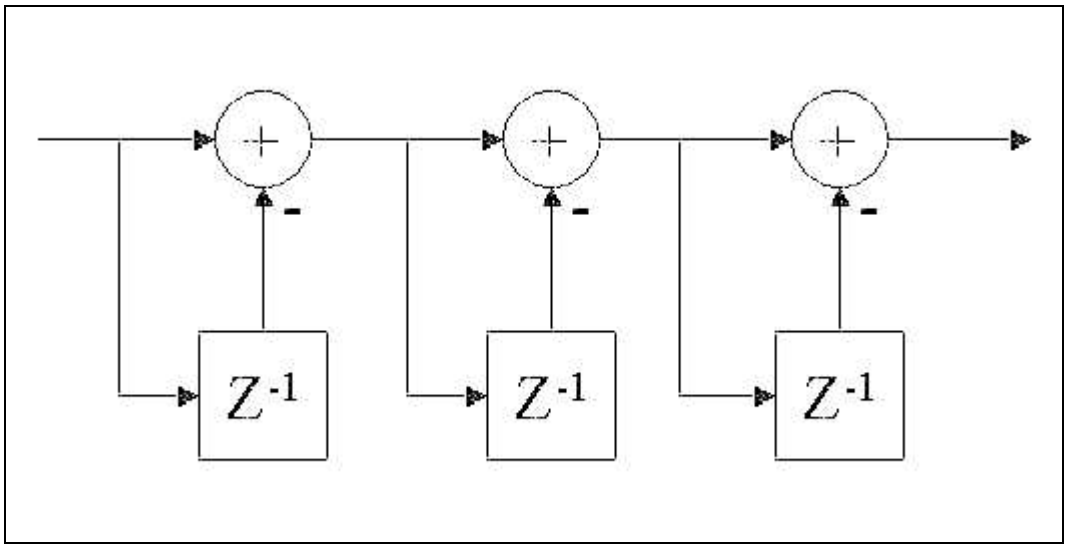


Figure 4: Block Diagram of Cascaded Comb Filter

Gexin

31.41: Re-sketch Fig.31.53 if, in each transfer function H(Z), a pole is added at DC.

Solution for 31.41: The zero at DC will be canceled due to the addition of pole at DC.

The magnitude response of the transfer function is

$$\left|H(Z)=\frac{\Pr oduct\_of\_dis\tan ces\_to\_zeros}{\Pr oduct\_of\_dis\tan ces\_to\_poles}\right|$$

For K=3:

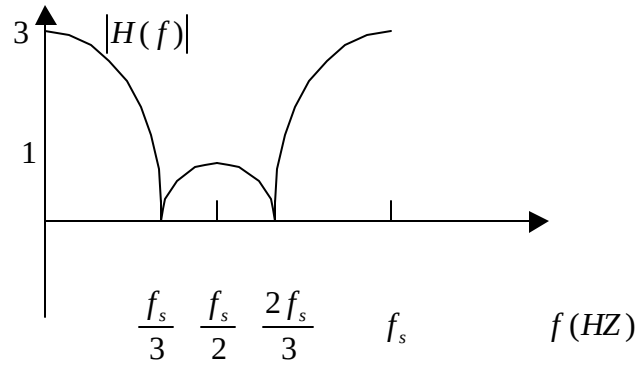
At DC point, the distances from DC point to two zeros are all equal to $\sqrt{3}$. While the distance from DC point to pole is equal to 1.

We get $|H(Z)|=3$.

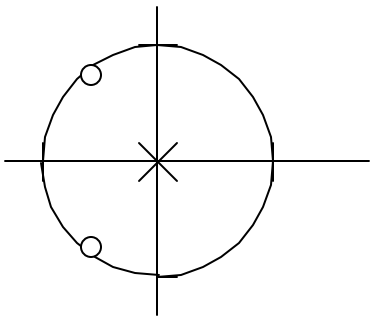
At $f_s/3$ and $2f_s/3$ points, ie. zero points, $|H(Z)|=0$.

At $f_s/2$ point, the distances to zeros are all equal to 1. We get $|H(Z)|=1$.

The frequency response for K=3 is shown below:



The Z-plane plot for zeros and poles is shown below:



For K=4:

At *DC* point, the distances from *DC* point to three zeros are equal to $\sqrt{2}$, $\sqrt{2}$ and 2 respectively. While the distance to pole is equal to 1.

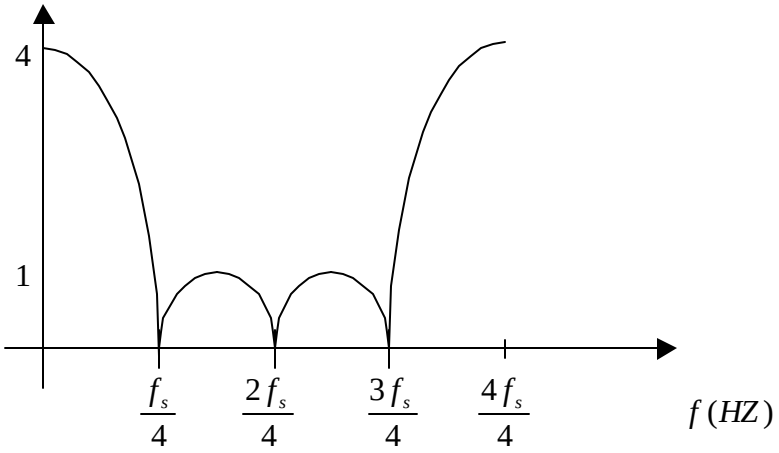
Thus $|H(Z)| = 4$ for DC.

At $f_s/4$, $2f_s/4$ and $3f_s/4$ points, ie. zero points, $|H(Z)| = 0$.

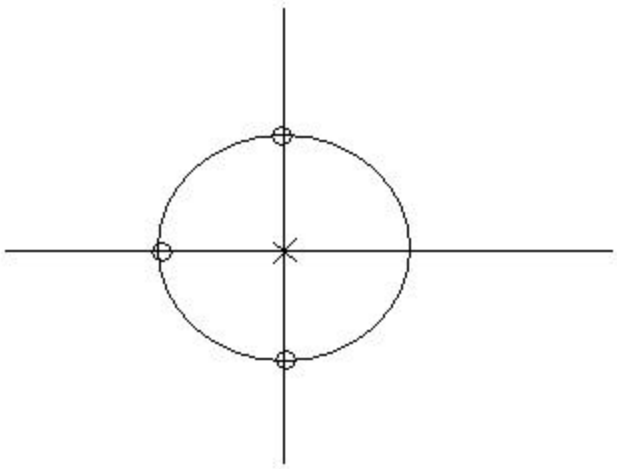
The peak amplitude value for the side lobe is $\left(2-\sqrt{2}\right)\cdot\left(\sqrt{2+\sqrt{2}}\right)\approx 1$

The frequency response plot for K=4 is shown below:

$|H(f)|$ $\qquad\qquad$ $H(Z) = \dfrac{1-Z^{-3}}{Z-1}$



Here is the Z-plane plot for poles and zeros.



For K=16:

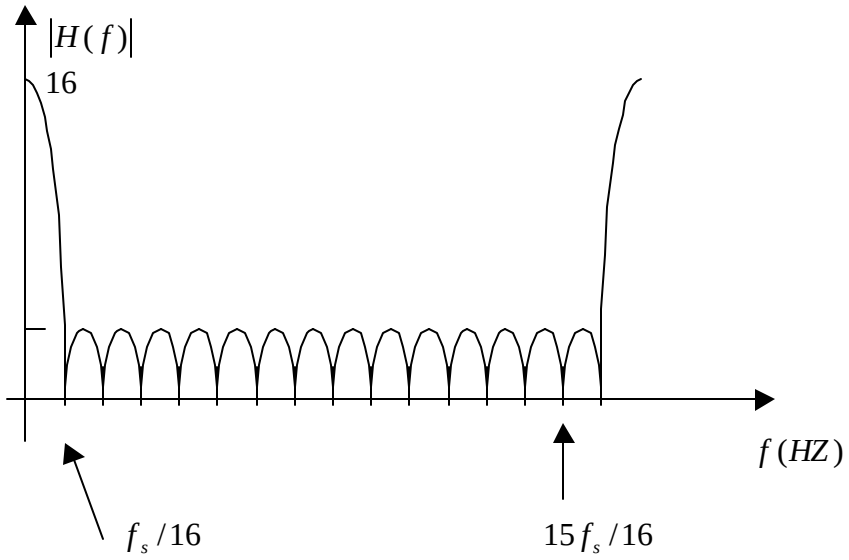
The transfer function is $H(Z) = \frac{Z^{16}-1}{Z^{16}} \cdot \frac{1}{Z-1}$

At *DC* point, $Z = 1$. The magnitude is given as follows:

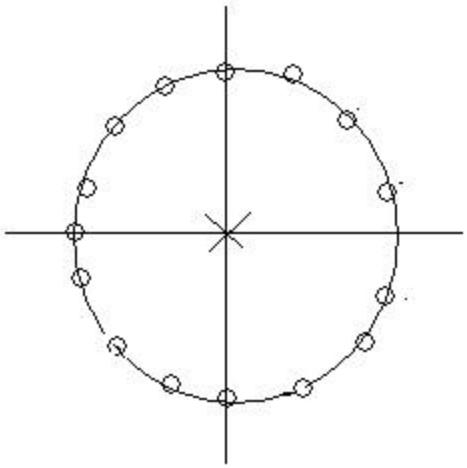
$|H(Z)| = \lim_{Z \to 1} (\frac{Z^{16}-1}{Z^{16}} \cdot \frac{1}{Z-1}) = 16$.

At $f_s/16$, $2f_s/16$ and $\ldots 15f_s/16$ points , ie. the zero points, the $|H(Z)| = 0$.

The frequency response plot for K=16 is shown below:



The Z-plane plot for poles and zeros is shown below:



Brandon Roth
brandonroth@micron.com

31.42 Show the problem with not using a MUX at the input of the adders in Fig. 31.55.

The MUX at the input of each integrator stage in Fig. 31.55 bit extends the input binary number to the same number of bits as the output of the integrator. This bit extending is necessary because the integrator output is feedback and added to the MUX output. When adding 2's complement binary numbers each number must have the same number of bits or the result will be incorrect.

For example, when adding a 6-bit and a 4-bit 2's complement binary numbers the result is incorrect without bit extending the 4-bit binary number.

x = 00 0111 = 7 (decimal)
y = 1000 = -8 (decimal)

Without bit extending
x + y = 00 0111
1000
00 1111 = 15 (2's complement)

With bit extending
x + y = 00 0111
11 1000
11 1111 = -1 (2's complement)

The bit extension is necessary to account for the sign of the 2's complement number especially when adding a positive and negative number.

Note that the first MUX in the filter also converts the output of the ADC from binary offset to 2's complement in addition to bit extending.

EE515: CMOS Mixed-Signal IC Design

Problem 31.43
Jim Slupe

31.43 Is it possible for the accumulate-and-dump circuit to output a spectrum with aliasing if the input signal is bandlimited to f_s? Why or why not?

Answer:

If the input signal has components as high as f_s then this clearly violates Nyquist sampling requirements, but that aside, energy is being put into all of the lobes (assuming an AAF on the input). In previous examples, even though lobes existed above B, we limited the energy to < B and so the lobes were, in effect, empty. Since resampling causes folding it is clear that energy above $(f_s/2)/K$ will be reflected into the baseband.

EE515: CMOS Mixed-Signal IC Design

Problem 31.44
Jim Slupe

31.44 In the discussions in this chapter we assumed the digital signals are much larger than an LSB of a data converter. What happens if this is not the situation for the Sync averaging filter?

Answer: The input to the Sinc filter appears as a DC signal if the ADC output is constant. The output of the ADC data converter, the input to the Sinc filter, will not vary if the input signal change is less than an LSB. If the input is varying a little amount so a few LSBs are changing then the quantization noise isn't going to be white and the result of passing the data through the Sinc filter may not cause a reduction in quantization noise.*

*The answer comes from Professor Baker. I was confused by the question and had to ask Professor Baker for help (this one is an easy ‘y’ for the checker).

EE515: CMOS Mixed-Signal IC Design
Brian Bergeson
Problem 31.45
bbergeso@poci.amis.com

Question 31.45:

Is it possible to decimate a digital waveform down to 2B and then later, with some other hardware or software digital filter, remove all of the aliased signal form the desired signal?

Answer:

No, you could not remove the aliased signal from the desired signal. When decimating down to 2B the aliased spectrum overlaps the desired spectrum (see Fig. 31.58b). Once this overlapping occurs there is no way to distinguish the aliased spectral content in 0 to B from the desired.

31.46 Suppose the waveform shown in Fig. 31.66 is the input to a decimator. If K = 8, what would the output of the decimator look like? Use integers to illustrate your understanding.

Assume the pattern of inputs in Fig. 31.66 is of the pattern: 10,11,12,13,14,15,16,17,15,13,11,9,7,5,3,1. Also assume that the initial output is 0 and that the pattern is clocked into a decimator at fs. The decimator adds up K samples, divides the sum by K (by truncating the log {base 2} K least significant bits), and outputs the result at fs/K. The output of the decimator will be 0 (initial output), 13 ([10+11+12+13+14+15+16+17]/8), and 8 ([15+13+11+9+7+5+3+1]/8).

EE515: CMOS Mixed-Signal IC Design
Question 31.47
Richard Friel
Rich_Friel@AMIS.COM

Question: #31.47

Suppose a digital word is clocked into a hold register and held for 8 clock cycles before another word is clocked into the register. Is this similar to the analog sample and hold? If the sampling rate (clock frequency) is increased by a factor of 8 after the hold register, what kind of digital filter can we think of the hold register as being?

Solution:

When an analog signal is sampled, the spectral response is equal to the Fourier Transform of the time domain representation of the input signal multiplied by the time domain representation of the sampling signal. An example is shown in (Eq. 30.16) and Fig. 30.27 and Fig. 30.28 in the text. This results in a Sinc function spectral response. In addition, the input signal spectrum becomes periodic with respect to the sampling signal spectrum. Aliasing is a concern when an analog signal is sampled.

When a digital word is clocked into a hold register for 8 clock cycles, this is similar to sampling and holding the digital input and the same spectral analysis and response is employed as when sampling and holding an analog signal. The spectral response of the input signal becomes periodic with respect to the spectral response of the sampling signal as shown in Fig. 31.69, node (b) and Fig. 31.70(b) in the text. The clock frequency band must be as least 2B of the input band, B, in order for aliasing to be minimized.

If the sampling rate (clock frequency) is increased by a factor of 8 after the hold register, this is similar to the first stage input of an interpolation digital filter as shown in Fig. 31.71, with the divide by 4 block replaced by a divide by 8 block. The spectral response of the input holding register in the digital interpolation filter is shown in Eq. 31.117 through Eq. 31.119, with the 4 replaced by a value of 8.

31.48 Show that the digital resonator of Fig. 31.76 can be modified, if we add a multiplier to the circuit, so that Eq. (31.120) can be implemented.

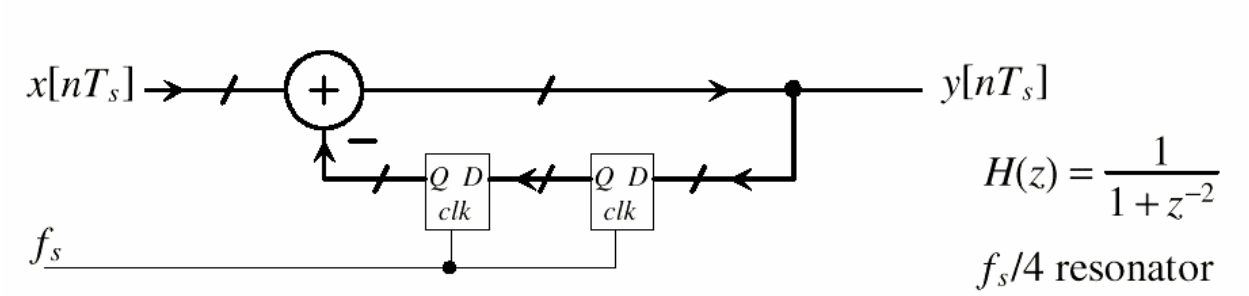
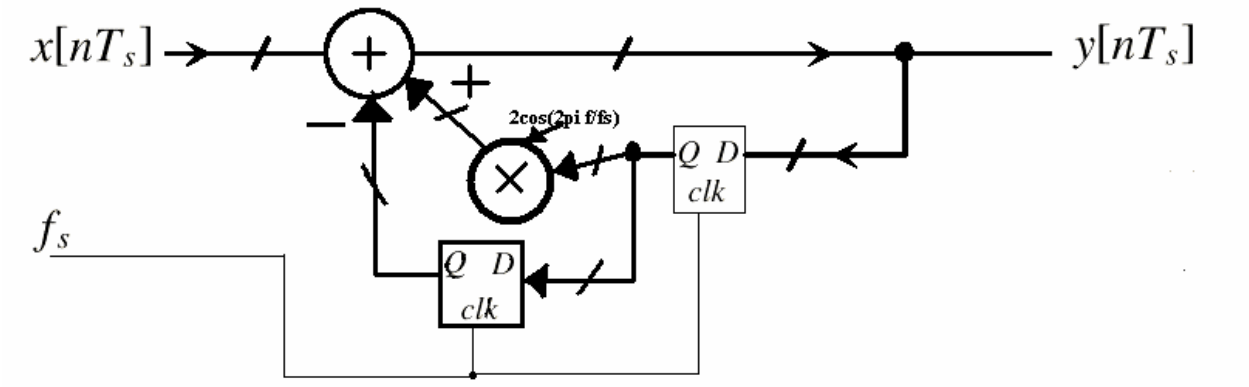


Figure 31.76 Implementation of a digital resonator.

Solution:

Given Eq.(31.120), $H_D(Z) = \dfrac{1}{1-2\cos[2\pi\ f/f_s] \cdot z^{-1} + z^{-2}} = \dfrac{z^2}{z^2 - 2\cos[2\pi\ f/f_s] \cdot z^1 + 1}$

Fig. 31.76 already incorporates the equation $H(z) = \dfrac{1}{1 + z^{-2}}$. The missing cosine term is added in the figure below.



Bob Moehrke EE515 (bmoehrke@amis.com)

Question 31.49

It is more correct to write our DAI continuous time input signals in Fig. 31.78 as

$$v_1(t) + V_{CM} \text{ and } v_2(t) + V_{CM}$$

Knowing this rederive Eq. (31.130)

If the above substitutions are to be made, equation 31.126 becomes

$$Q_1 = C_I \left(V_{CM} - v_1 [(n-1)T_s] - V_{CM}\right)$$

and equation 31.127 becomes

$$Q_2 = C_I \left(V_{CM} - v_2 [(n-1/2)T_s] - V_{CM}\right)$$

As before, $Q_1 - Q_2$ is transferred to the feedback capacitor the next time ϕ_1 is closed. This creates a change in the output voltage V_{out} described by equation 31.128. It can easily be seen that the old quantity $Q_1 - Q_2$ and the new quantity $Q_1 - Q_2$ are identical. Equation 31.128 then remains the same. Equation 31.129 is the z-domain form of equation 31.128, and equation 31.130 is an algebraic version of 31.129. These last two equations (31.129 and 31.130) are the same as before as well.

31.50
Jeremy Rice

Using: $V_1 = V_1 + Vcm$

$V_2 = V_2 + Vcm$

Rederive Eq 31.13

$$Vout(z) := \frac{Ci}{Cf} \cdot \frac{V(z)_1 \cdot z^{\frac{1}{2}} - V(z)_2}{1 - z^{-1}}$$

Start with the charge on the capacitors,

$$Q1 := Ci \cdot \left[Vcm - \left[V\left[\left(n - \frac{1}{2} \right) Ts \right]_1 + Vcm \right] \right]$$

$$Q2 := Ci\left[Vcm - \left(V(n \cdot Ts)_2 + Vcm \right) \right]$$

From which, it is immediately obvious that the Vcm term drops out

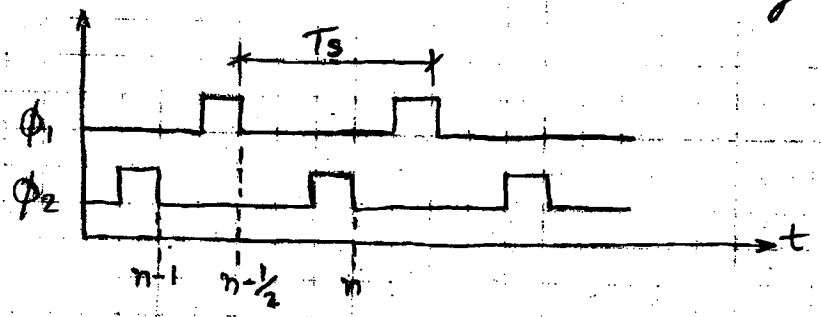
$$Vout(n \cdot Ts) - Vout[(n-1) \cdot Ts] \cdot Cf = Ci \cdot \left[-V(n \cdot Ts)_2 + V\left[\left(n - \frac{1}{2} \right) Ts \right]_1 \right]$$

$$Vout(z) \cdot \left(1 - z^1\right) = \frac{Ci}{Cf} \cdot \left(V_1 \cdot z^{\frac{-1}{2}} - V_2 \right)$$

$$Vout(z) = \frac{Ci}{Cf} \cdot \frac{V(z)_1 \cdot z^{\frac{-1}{2}} - V(z)_2}{1 - z^{-1}}$$

Which is the same as Eq. 31.1, so it can be said that common mode signals will be removed by the circuit.

Problem 31.51 Using the result from problem 31.49 derive the transfer function, Eq. (32.139), for the circuit shown in Fig. 32.92



Since there's no switch on the output of the op-amp in Fig. 32.92, Vout only changes states when ϕ_2 switches close. So Vout will change its states at either time $(n-1)T_S$ or nT_S.

For the bottom portion, when ϕ_1 closes,

$Q_{f1} = 2C_f\left(V_{CM} + V_{out}[(n-1)T_S]\right)$.

using the result from problem 31.49,

$Q_{f1} = 2C_f\left(V_{cm} + V_{out}[(n-1)T_S] - V_{CM}\right) = 2C_f \cdot V_{out}[(n-1)T_S]$.

$Q_{f2} = 0$, when ϕ_2 closes, since V_{CM} is applied to both side of the $2C_f$ switched capacitor. The charge transferred to the feedback capacitor C_f is $Q_{f2} - Q_{f1} = -2C_f \cdot V_{out}[(n-1)T_S]$.

For the top portion, when ϕ_1 closes,

$Q_{C1} = C_I\left(V_{CM} - V_1[(n-\tfrac{1}{2})T_S]\right)$

$Q_{C2} = C_I\left(V_{CM} - V_2[nT_S]\right)$ when ϕ_2 closes

$Q_{C2} - Q_{C1} = C_1\left(V_1[(n-\tfrac{1}{2})T_S] - V_2[nT_S]\right)$.

Since the C_I and $2C_f$ switched capacitors are in parallel, when the ϕ_2 switches are closed, the charge transferred from both the C_I and $2C_f$ can be added. This leads to

$$\left(V_{out}[nT_S] - V_{out}[(n-1)T_S]\right)\cdot C_f = C_I\left(V_1[(n-\tfrac{1}{2})T_S] - V_2[nT_S]\right) - 2C_f \cdot V_{out}[(n-1)T_S]$$

$$C_f\cdot\left(V_{out}[nT_S] - V_{out}[(n-1)T_S + 2V_{out}[(n-1)T_S]\right) = C_I\left(V_1[(n-\tfrac{1}{2})T_S] - V_2[nT_S]\right)$$

$$V_{out}[nT_S] + V_{out}[(n-1)T_S] = \frac{C_I}{C_f}\left(V_1[(n-\tfrac{1}{2})T_S] - V_2[nT_S]\right)$$

in z-domain

$$V_{out}(z)\left[1 + z^{-1}\right] = \frac{C_I}{C_f}\left(V_1(z)\cdot z^{-1/2} - V_2(z)\right)$$

$$V_{out}(z) = \frac{C_I}{C_f}\cdot\frac{V_1(z)\cdot z^{-1/2} - V_2(z)}{1 + z^{-1}}$$

Tyler J. Gomm
tjgomm@micron.com

31.52 Show the detailed derivation of Eq. (31.138).

Eq. (31.138) relates the inputs to the outputs of a feedback modulator, where quantization noise is represented in the *z*-domain by E(z).

$$Y(z) = \frac{A(z)}{1 + A(z) \cdot B(z)} \cdot X(z) + \frac{1}{1 + A(Z) \cdot B(z)} \cdot E(z) \qquad (31.138)$$

Figure 1 (similar to Fig. 31.82) shows the block diagram of a feedback modulator.

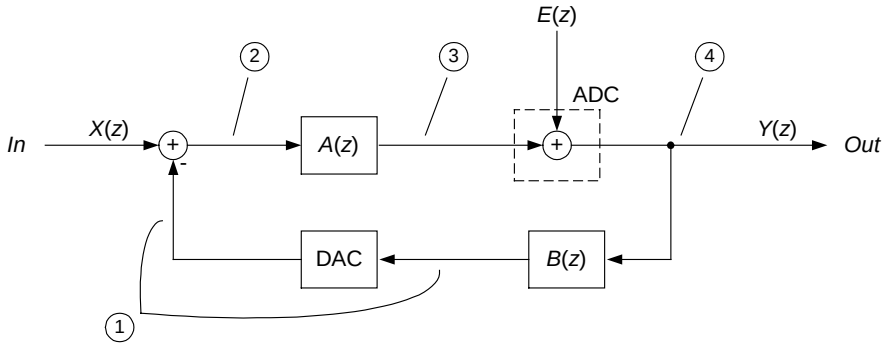


Figure 1 Block diagram of a feedback modulator.

To derive the transfer function of Eq. (31.138), we must examine the values at nodes 1, 2, 3, and 4 by going around the loop as follows:

$$(1) = Y(z)B(z)$$
$$(2) = X(z) - Y(z)B(z)$$
$$(3) = X(z)A(z) - Y(z)A(z)B(z)$$
$$(4) = X(z)A(z) - Y(z)A(z)B(z) + E(z)$$

Because node 4 is also $Y(z)$, we can equate the above to $Y(z)$ and solve.

$$Y(z) = X(z)A(z) - Y(z)A(z)B(z) + E(z)$$

$$Y(z) + Y(z)A(z)B(z) = X(z)A(z) + E(z)$$

$$Y(z)\left[1 + A(z)B(z)\right] = X(z)A(z) + E(z)$$

$$Y(z) = \frac{1}{1 + A(z)B(z)}\left[X(z)A(z) + E(z)\right]$$

$$Y(z) = \frac{A(z)}{1 + A(z)B(z)}X(z) + \frac{1}{1 + A(z)B(z)}E(z) \qquad (31.138)$$

Tyler J. Gomm
tjgomm@micron.com

31.53 Summarize the advantages and disadvantages of predictive and noise-shaping data converters.

Both predictive and noise-shaping modulators use oversampling, which increases the final converter resolution.

A predictive modulator attempts to feedback an analog signal with the *same* value of the input signal. This drives the output of the summer (see Fig. 31.81) toward zero, reducing the required input range of the ADC. However, in order to recover the input signal, the output of the predictive modulator must be passed through an analog filter with a transfer function identical to that of the feedback filter used in the modulator. This required precision of the analog components directly limits the attainable resolution. Another disadvantage is that the predictive modulator shapes *both* the input signal spectrum as well as the quantization noise.

A noise-shaping modulator feeds back the *average* value of the input, which can be filtered to reduce the accuracy required of the analog components. Because of this filtering (averaging), the analog components in the forward path (Fig. 31.81) require less accuracy. However, the output of the DAC in the feedback path does not experience this averaging, and therefore, the DAC must be linear to the required accuracy of the entire converter. No precision filter or analog components are required (other than the DAC). Another advantage is that the signal transfer function (STF) approaches unity while the noise transfer function (NTF) is shaped such that the energy in the spectrum is moved out of the bandwidth of interest.