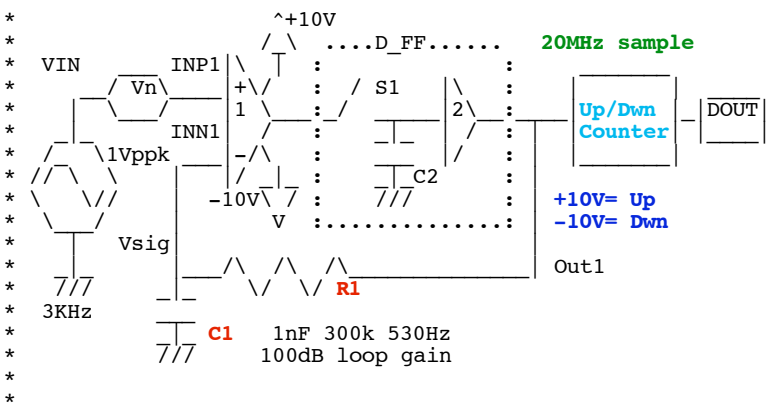
Re-Inventing Delta Modulation

**Back in the late 70s and early 80s, it was believed at National Semiconductor,
that it was impossible to build a Analog to digital convertor
with greater than 12bits.**

**The argument was that the temperature coefficient of the matching elements
was not low enough.**

**The following idea was presented to Rod Russell,
who was the design manager for ADCs and DACs at that time.
The argument was, if you did this, why can't you build a 20bit ADC?**

How to build an ADC using a Counter, a Comparator, and a R and C.



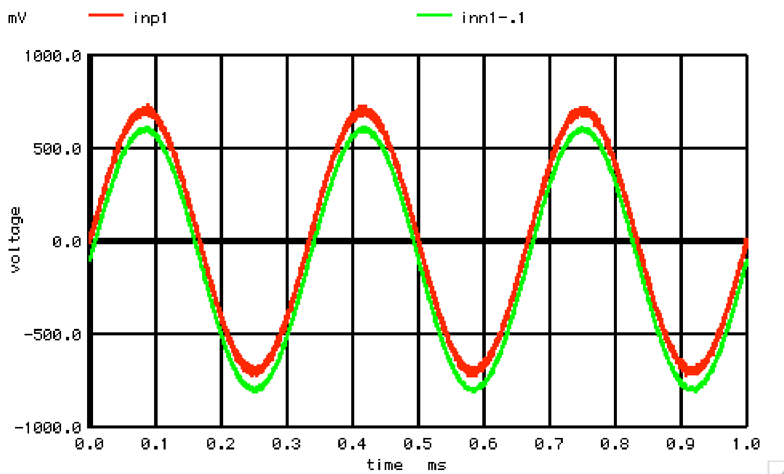
**At the time, Opamps were designed for +/-15V supplies.
Take the fastest voltage comparator with a latching output.
Such a comparator always has a large referred to input noise,
since its bandwidth is as high as possible.**

**It should be possible to find a set of Resistor/Capacitor feedback values
where the comparator is stable in negative feedback.**

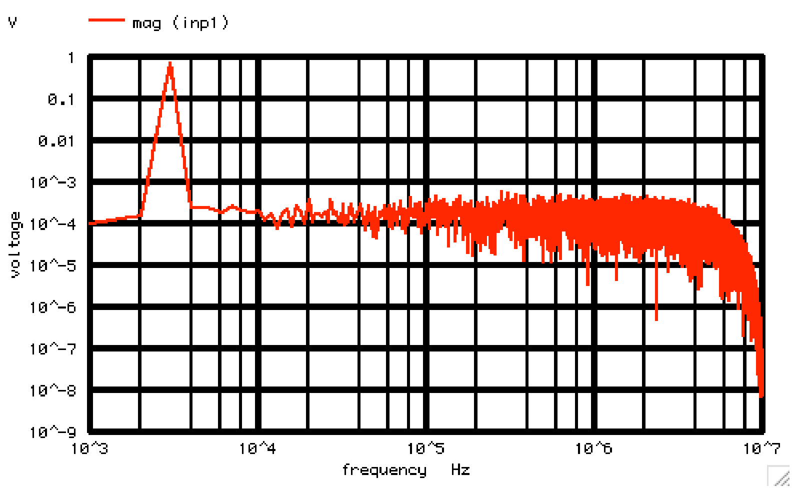
**Now if the input signal is a slow sine wave,
the voltage across the capacitor will track it.
Since the resistor/capacitor is being driven by a latched digital signal,
how is the value of the voltage across the capacitor
operating any different from that of an Up/Down counter
which is connected to the same latched digital output node?**

**As it turns out, this Delta Modulation method was invented in 1946.
And at that very same time, Analog engineers were rediscovering it.
But nobody at National Semiconductor had seen this concept it before.
It was not long afterwards that Sigma-Delta ADCs became mainstream.**

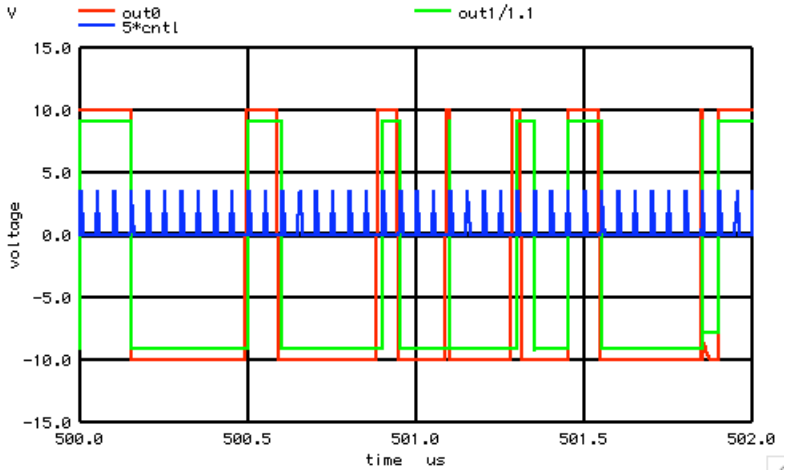
```
*
*                   ^+10V
*                  /_\   ....D_FF......    20MHz sample
*  VIN      ___ INP1|\   |   :              :
*       __/ Vn\____|+\/   :  / S1   |\     :   _______
*      |  \___/    |1 \___:_/  _____ |2\___:____|Up/Dwn  |_|DOUT|
*     _|_      INN1|   /   :    _|_   |  /  :  |Counter |  |____|
*    /   \1Vppk ___|-/\    :    ___   | /   :  |        |
*  // \  \      |   |/ _|_ :    _|_C2       :  |________|
*  \    \//     |  -10V\ / :    ///         :  +10V= Up
*   \___/       |      V   :................: -10V= Dwn
*     |    Vsig |                            |
*    _|_        |___/\  /\  /\______________| Out1
*    ///       _|_    \/  \/ R1
*   3KHz
*              ___ C1   1nF 300k 530Hz
*              ///      100dB loop gain
*
*
```



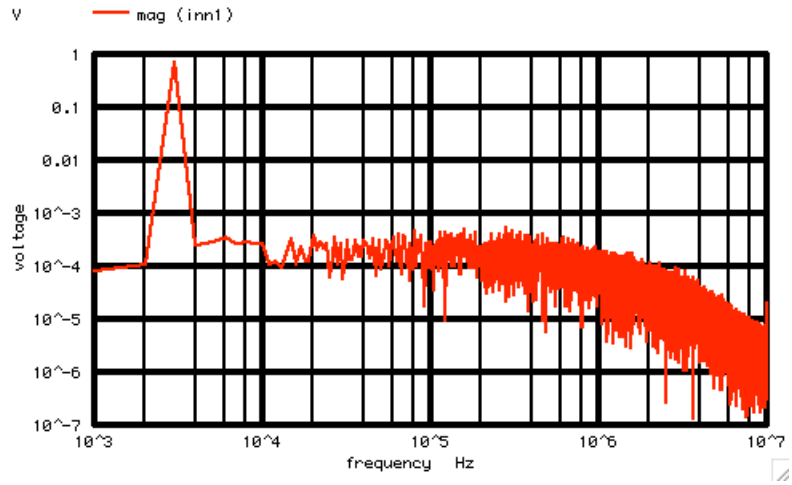
The plots of the two inputs to the comparator are shown above.
The negative input voltage has an offset of 100mV to show both inputs.
A 5Mhz random PWL noise voltage has been put in series with the input signal to simulate the input noise of the comparator.



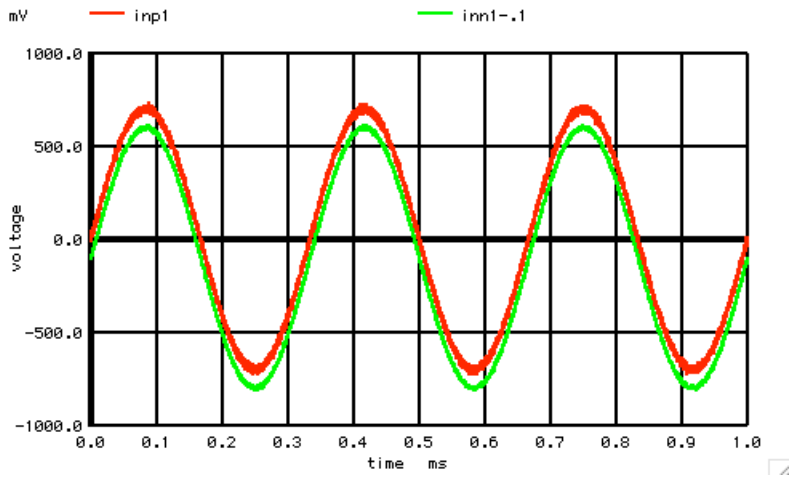
The input spectrum that the comparator sees is shown above.
The noise bandwidth was limited to 5Mhz to save on simulation time.



The output of the comparator is being latched at a 20MHz rate.
This simulation is using a simple sample and hold.



The spectrum of the capacitor voltage will have a slightly lower bandwidth.
At lower frequencies, the capacitor will track the input signal,
noise and all.



The voltage across the capacitor shows what the a UP/Down counter
running off the same latch digital output should be seeing.

There appears to be less noise in the capacitor,
since the comparator was not simulated with infinite gain bandwidth.

But for frequencies below 100kHz, the spectrums are effectively the same.

Now to create high bandwidth comparators,
input stages need to run a lot of supply current.
Higher input currents lower impedances and input noise voltages.
While the comparator's bandwidth will make the referred to input noise large,
it is only it's noise at the lower frequencies that count.

```
===================MacSpiceCode===========================================
ShownRodRussell
*                                                    R3
*
*                    ^+10V
*                   /_\  ....D_FF......   20MHz sample
*  VIN    ____ INP1|\  |  :             :    _______
*      __/ Vn\____|+\/   :  / S1   |\   :   |       |  ____
*     |  \____/    |1 \___:_/  _____|2\__:___|Up/Dwn |_|DOUT|
*    _|_       INN1|  /   :   _|_   | /  : | |Counter| |____|
*   /_  \1Vppk ____|-/\   :   ___   |/   : | |_______|
*  // \  \    |    |/ _|_ :   _|_C2      : |
*  \   \//    |  -10V\ /  :   ///        : |  +10V= Up
*   \___/     |      V    :..............: |  -10V= Dwn
*     |  Vsig |                            |
*     |       |___/\  /\  /\_______________|  VSQ
*    ///     _|_     \/  \/ R1
*   3KHz     ___
*            _|_ C1   1nF 300k 530Hz
*            ///     100dB loop gain
*
*

*================Need_A_voltage_Source_to_alter====================="
*V_PULSE#        NODE_P NODE_N DC     VALUE  PULSE( VINIT  VPULSE TDELAY TRISE  TFALL  PWIDTH PERIOD )
VCNTL            CNTL   0      DC     0      PULSE( 0      1      .5p     1n     1n     5n     50n    )
*V_SIN#          NODE_P NODE_N DC     VALUE  SIN(   V_DC   AC_MAG FREQ    DELAY  FDamp)
VIN              VIN    0      DC     0      SIN(   0      .7      3k     )
VN1              INP1   VIN    DC     0

R1               INN1   OUT1   300k
C1               INN1   0      1n
C2               OUT00  0      100p
B1               OUT0   0      V =  10*tanh(100*tanh(100*tanh(100*tanh(100*tanh( V(INP1) -V(INN1))))))
BB               OUT1   0      V  =  V(OUT00)

*S_NUMB          NODE1  NODE2  CNTL_P CNTL_N MODEL  ON/OFF
S1               OUT0   OUT00   CNTL   0      SW

.MODEL           SW     SW(    VT=.2 VH=.1m  RON=10m  ROFF=1MEG)

.control
set              pensize = 2
echo             "=======================Want_10000_.1us_steps=============================="
let n =          10000
let tstep =      .1us
let period_t =   n*tstep
let Bin_Hz =     1/period_t
let nyquist =    .5/tstep
echo             "Total_Period_s =      $&period_t"
echo             "Bin_Resolutio_Hz =    $&Bin_Hz"
echo             "Sample_Period_s =     $&tstep"
echo             "Nyquist_Hz =          $&nyquist"
echo             "=======================Create_PWL_array_and_Index_and_Plot=============="
let pwl_1 =      vector(2*n)*tstep*0.5
let ii =         vector(2*$&n)
*plot            pwl_1  vs ii
echo             "=======================Add_1Vrms_Noise_to_PWL_array====================="
let index =      0
repeat           $&n
let              pwl_1[1+2*index] = 10m*(rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+
rnd(127)-507.5)/102.879
let index =      index + 1
end
echo             "=======================Install_the_PWL_array============================"
alter            @vn1[pwl] = pwl_1
echo             "=======================Run_and_Plot====================================="
let period_s =   tstep/2
let trans_per =  tstep/20
tran             $&trans_per $&period_t   0  $&trans_per
*echo             "run here"
plot             inp1 inn1-.1
```

```
*plot             inp1 out1 inn1 out0
plot             out1 inn1
plot             out0   out1/1.1  5*cntl xlimit .5m .502m

echo             "======================FFT_and_Plot====================================="
linearize
let              FFT_BandWidth_Hz =      10Meg
let              FFT_resolution_Hz =     1k
echo             "FFT_BandWidth_Hz=      $&FFT_BandWidth_Hz"
echo             "FFT_resolution_Hz=     $&FFT_resolution_Hz"
set              specwindow=             "rectangular"
spec             $&FFT_resolution_Hz     $&FFT_BandWidth_Hz   $&FFT_resolution_Hz     v(inn1)
let expect_V =   (sqrt(2)/sqrt(500k/1k))/(1+(frequency/550k)*(frequency/500k)*(frequency/500k)*(frequency/
500k)*(frequency/500k))
plot             mag (inn1) loglog

destroy
destroy

echo             "======================FFT_and_Plot====================================="
linearize
let              FFT_BandWidth_Hz =      10Meg
let              FFT_resolution_Hz =     1k
echo             "FFT_BandWidth_Hz=      $&FFT_BandWidth_Hz"
echo             "FFT_resolution_Hz=     $&FFT_resolution_Hz"
set              specwindow=             "rectangular"
spec             $&FFT_resolution_Hz     $&FFT_BandWidth_Hz   $&FFT_resolution_Hz     v(inp1)
let expect_V =   (sqrt(2)/sqrt(500k/1k))/(1+(frequency/550k)*(frequency/500k)*(frequency/500k)*(frequency/
500k)*(frequency/500k))
plot             mag (inp1) loglog
echo             "==================done============================================="


.endc
.end


7.22.11_12.49PM
dsauersanjose@aol.com
Don Sauer
```