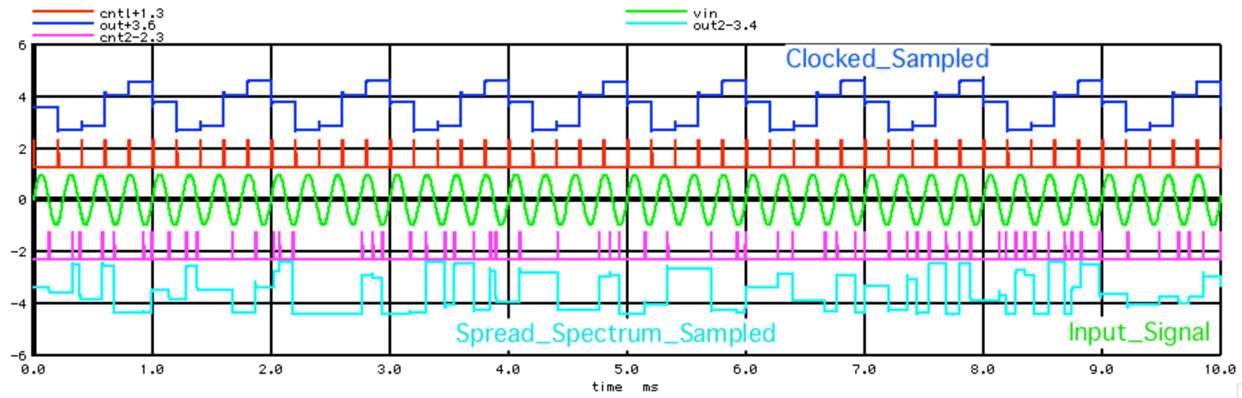
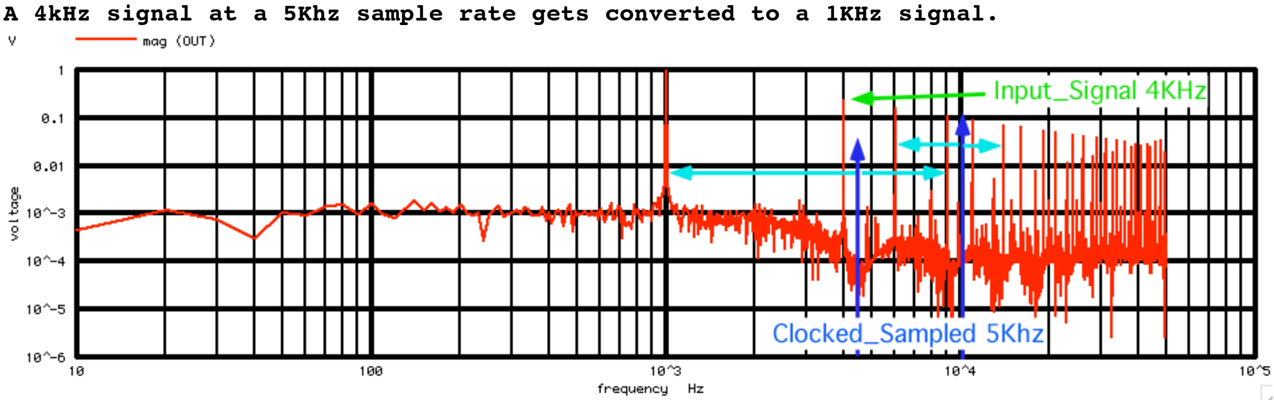
====================Aliased_Spread_Spectrum_Sampling==================================

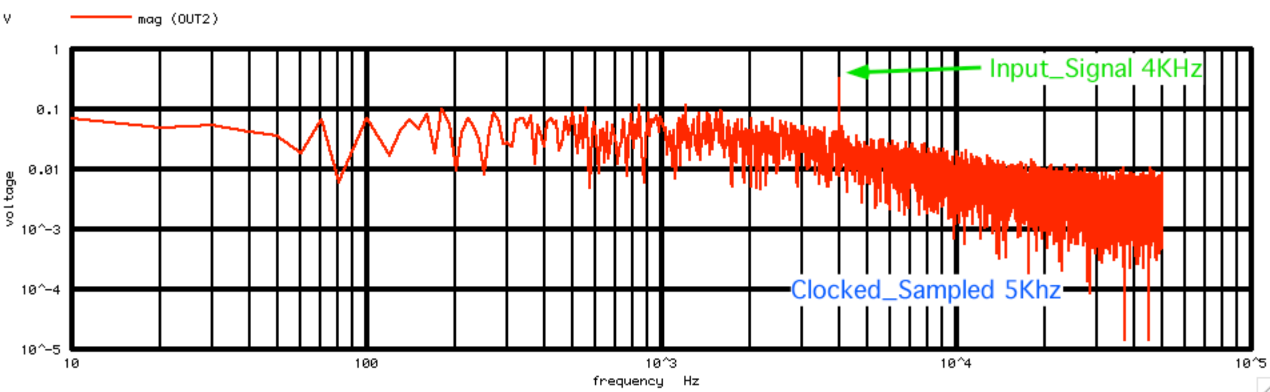
1) For Clocked ADC the vin signal generate harmonics above and below multiples of
   the sample frequency.
2) Signals above Nyquist get aliased as normal signal.
3) Spread Spectrum Sampling at +/- 1_rms radian of Phase Modulation can stop this.



A 4kHz signal at a 5Khz sample rate gets converted to a 1KHz signal.



A 4kHz signal input to a 5Khz sample rate causes the harmonics to be mixed everywhere.



A spread sprectrum sampling spreads all the harmonic evenly to look like noise.

==================MacSpiceCode==

Aliased_Spread_Spectrum_Sampling

```
*=========D_FF=========================================
*
*              IN2
*            |\    /           IN3  |\
*    |IN |_|1\__/  ____________ |2\____|OUT|
*    |___|  |  / S1      _|_      |  /     |___|
*            |/                   |/
*                  C1 ___
*                       |
*                      ///
*===============Need_voltage_Sources_to_alter_with_PWL_Data===================="
VT      Vtime      0       dc    0      PWL  ( 0  0  1  1 )
B0      SAW        0       V =          atan(tan(3.141592653589793*500*v(Vtime)))
V1      V1         0       dc    0
*VSIN   NODE_P     NODE_N  DC    VALUE  SIN(    V_DC    AC_MAG FREQ    DELAY   FDamp)
VIN     VIN        0       dc    0      SIN(    0       1      4000          )
BVref   Vref       0       V =          u(sin(6.2831*5000*V(VTime)))
BVjit   Vjit       0       V =          u(sin(6.28*5000*V(Vtime)+V(V1)))
```

```
XPE1   Vref      CNTL    POS_E
XPE2   Vjit      CNT2    POS_E
XS_H1  VIN       CNTL    OUT      SH
XS_H2  VIN       CNT2    OUT2     SH

.control
set              pensize = 2
echo             "==================Want_1000_.1ms_steps======================="
let n =          1000
let Nlev =       127
let tstep =      .1ms
let Nrnd =       8
let Nbins =      Nlev*Nrnd
echo             "random levels      0-> $&Nlev"
echo             "Numb rnd waveforms $&Nrnd"
echo             "==================Create_PHaseNoise_array=================="
let PNoise =     vector($&n)
let IntPNoise =  vector($&n)
let ii =         vector($&n)
let index =      0
repeat           $&n
let              PNoise[index] = 3.14*(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
*plot             PNoise  vs ii
let averVal =    mean(PNoise)
let noisAC =     PNoise - averVal
let RmsVal =     sqrt(mean(noisAC* noisAC))
echo             "Average level      $&averVal"
echo             "RMS level          $&RmsVal"
echo             "==================Create_PWL_arrays========================="
let pwl_1 =      vector(2*n)*.5*tstep
echo             "==================Create_Integrated_PHaseNoise_array==========="
let n2 =         n
let index =      0
repeat           $&n2
let              pwl_1[2*index+1] = PNoise[index]
let index =      index + 1
end
echo             "==================Install_the_PWL_arrays========================="
alter            @v1[pwl] = pwl_1
echo             "==================Run_and_Plot===================================="
tran             10u   100m   0  10u
plot             cntl+1.3 vin out+3.6 out2-3.4  cnt2-2.3 xlimit 0 10ms

echo             "==================FFT_and_Plot_OUT===================================="
linearize
let              FFT_BandWidth_Hz =  50K
let              FFT_resolution_Hz = 10
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(OUT)
plot             mag (OUT)   loglog
*plot            mag (OUT)  ylog xlimit 400 600
echo             "==================FFT_and_Plot_OUT2===================================="
destroy
let              FFT_BandWidth_Hz =  50K
let              FFT_resolution_Hz = 10
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(OUT2)
plot             mag (OUT2)   loglog
echo             "==================Done=========================================="

.endc

*=========Sample_Hold============================================
*
*            IN2
*     ___  |\    /         IN3 |\     ___
*    |IN |_|1\__/  ___________ |2\____|OUT|
*    |___| |  / S1   _|_       |  /   |___|
*          |/                  |/
*                 C1 ___
*                     |
*                    _|_
*                    ///
.SUBCKT   SH     IN     CNTL      OUT
 B1       IN2    0      V =       v(IN )
 S1       IN2    IN3    CNTL     0    SW
 C1       IN3    0      .1u

 R1       IN3    0      10Meg
 B2       OUT    0      V =       v(IN3 )
.ENDS     SH
*=========POS_Edge===========================================
*
*        |\              VBF ____              ___
* |IN |__|3\_________________|    \             |   \
* |___|  | /  |           VLP|  PE \_|OUT|  |IN |_|PE  \_|OUT|
*        |/   |/\  /\  /\____|     / |___|  |___| |    / |___|
*                 \/  \/   | |____/               |__/
*               RLP       _|_
*                         ___
*                    CLP   |
*                         _|_
*                         ///
*
.SUBCKT   POS_E  IN     OUT
 BBUF     VBF    0      V =   u( v(IN )-.5 )
```

```
 RLP      VBF    VLP    10k
 CLP      VLP    0      1n   IC=0
 BAND     OUT    0      V =    u( u(v(VBF )-.5)*u(.5 -v(VLP ) ) -.1)
.ENDS     POS_E

.MODEL    SW      SW(    VT=.5 VH=.1  RON=1 ROFF=100MEG)
.end
```

4.4.11_12.26PM

dsauersanjose@aol.com
Don Sauer