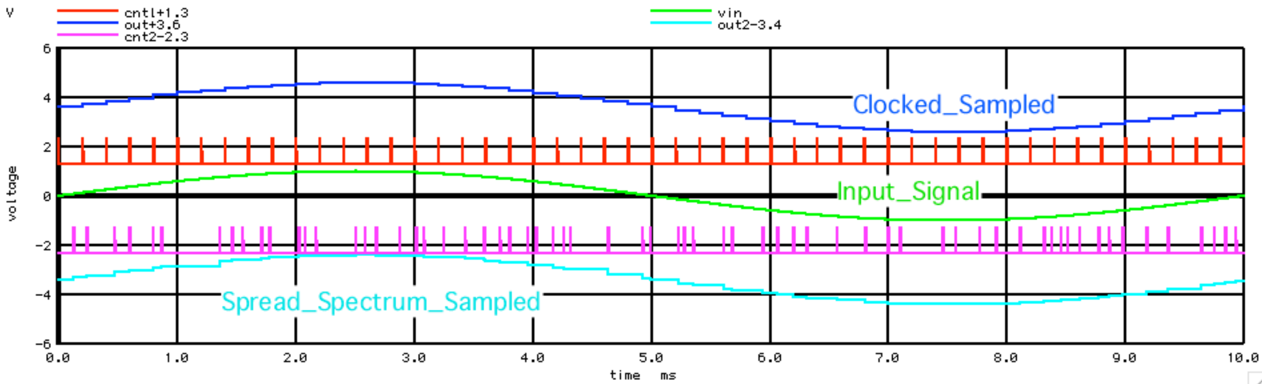
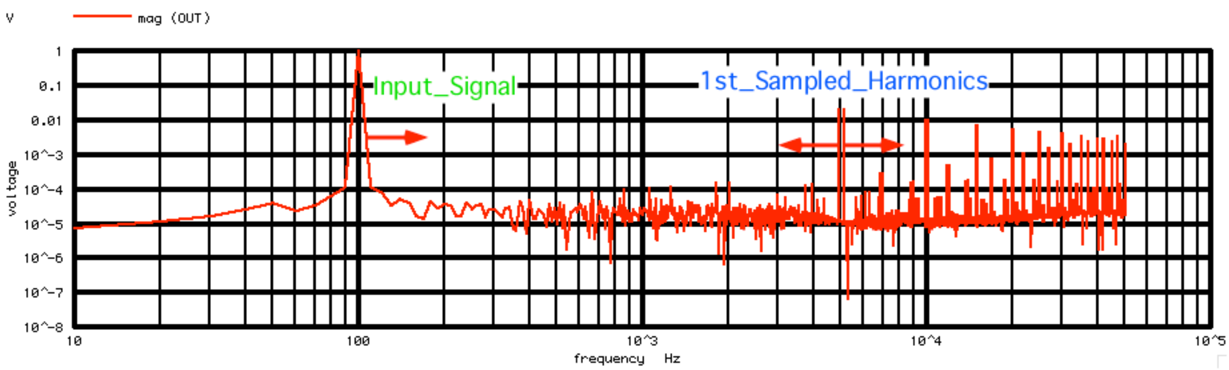
=====================Spread_Spectrum_Sampling=====================================

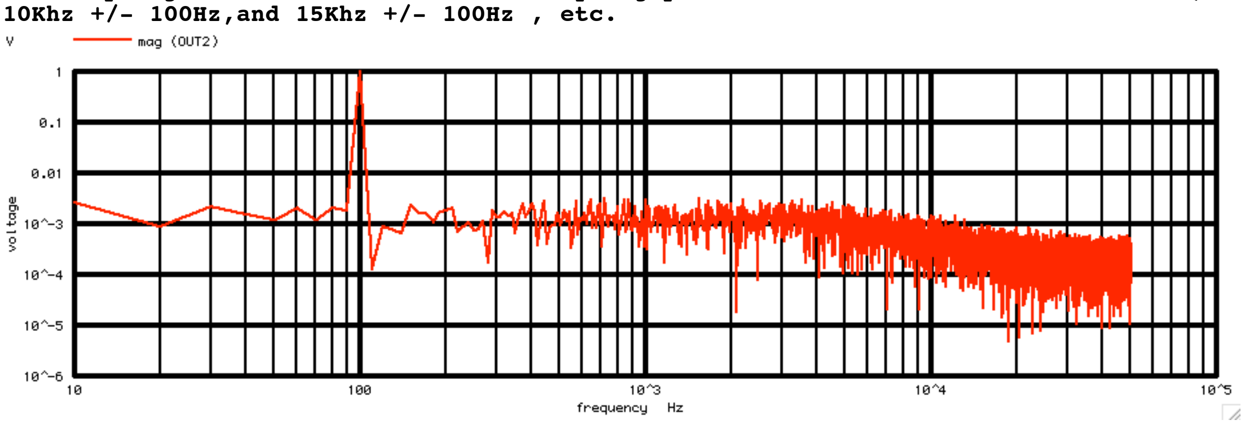
1) For **Clocked ADC**, the vin signal generates **harmonics above and below multiples** of the sample frequency.
2) Adding around **+/- 1_rms radian of Phase Modulation to the clock** appears to completely spread these harmonics out over the entire spectrum.
3) A **paper** on Spread Spectrum Sampling is **included below.**



A sub Nyquist frequencies, the clocked and spread spectrum samples look about the same.



The sampling is at 5Khz. So a clock sampling produces harmonics at 5Khz +/- 100Hz, and 10Khz +/- 100Hz,and 15Khz +/- 100Hz , etc.



What the spread spectrum sampling does it that it spreads all the harmonic evenly around the whole spectrum.

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

Spread_Spectrum_Sampling

```
*
*              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     100       )
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
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
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_12.22PM

dsauersanjose@aol.com
Don Sauer