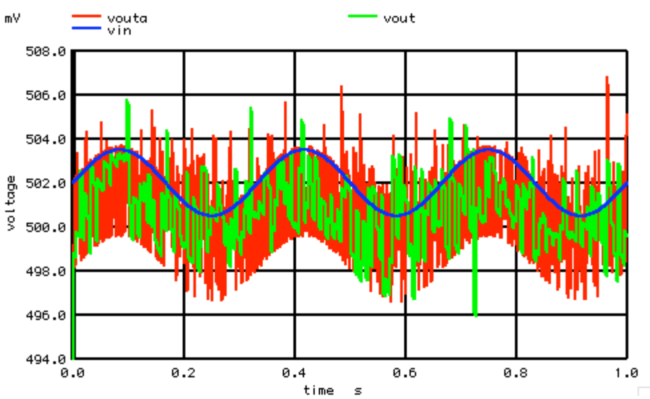
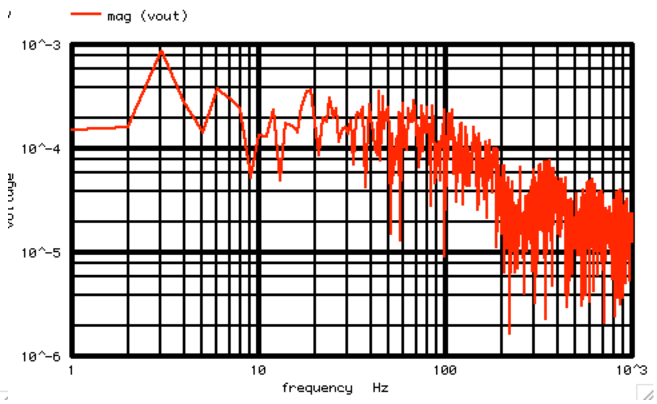
=====================Dithering====================================

1) The human ear has a dynamic range from 0dB (which is the noise level of air)
   to 196dB (one atmosphere = destructive!)
2) Digital audio first came out at 16Bits (which is 96dBs).
3) Signal at the LSB level does develop nasty harmonics which mix with other signal.
4) The Dithering technique was applied early on.
5) The addition of dither noise in the LSB range can spread spectrum the nasty LSB harmonics.
6) In fact the Dithering technique can allow detection below the LSB level.





Here a input signal is applied which is below an LSB. The addition of noise
adds a probablity factor to a sub LSB signal.

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

```
Adding_Dither
*
*       VIN
*
*    __|_________|_THO____|_    _TH1_____|_    _THO_____|_    _TH1_____|_    _THO_____|_    _TH1_____|_    _THO_____|_       TH1
*   / \         |  |\   _/ \_  |  |\  _/ \_  |  |\  _/ \_  |  |\  _/ \_  |  |\  _/ \_  |  |\  _/ \_  |  |\  _/ \_      |\
*  //\ \        |  | \  \___/  |  | \ \___/  |  | \ \___/  |  | \ \___/  |  | \ \___/  |  | \ \___/  |  | \ \___/     | \
*  \  \//       |_ |  \_  |   _|  |  \_  |  _|  |  \_  |  _|  |  \_  |  _|  |  \_  |  _|  |  \_  |  _|  |  \_  |______|  \_
*   \__/           |  /   OFF0    |  /   OFF1   |  /   OFF2   |  /   OFF3   |  /   OFF4   |  /   OFF5   |  /   OFF6     |  /
*   __|__          | /            | /           | /           | /           | /           | /           | /              | /
*    ///
*                    _|_           _|_          _|_           _|_           _|_           _|_           _|_              _|_
*                   |D0 |         |D1 |        |D2 |         |D3 |         |D4 |         |D5 |         |D6 |            |D7 |
*                   |___|         |___|        |___|         |___|         |___|         |___|         |___|            |___|
*
*
*         _________
*        __|       \
*        __|        \
*        __|  DAC    \__ VOUTA
*        __|         /
*        __|        /
*        __|_______/
*
*
*   VOUTA      IN2
*    ___    |\    /          IN3  |\
*   |IN |_ _|1\__/   ____________ |2\_____|OUT|
*   |___|   |  / S1      _|_      |  /    |___|
*           | /                   | /
*               C1  |
*                  _|_
*                  ///
```

```
*=========Create_Signal==================
*V_SIN#   NODE_P NODE_N DC     VALUE  SIN(   V_DC   AC_MAG FREQ    DELAY   FDamp)
VIN       VIN    0      DC     0      SIN(   .502   .0015   3                     )
BVIN2     VIN2   0      V =    V(VIN) + V(V1)
*V_PULSE# NODE_P NODE_N DC     VALUE  PULSE( VINIT  VPULSE TDELAY TRISE  TFALL  PWIDTH PERIOD )
VCLKL     CLK    0      DC     0      PULSE( 0      1      1n     1n     1n     2m     4m     )
V1        V1     0      DC     0

XPOSE1    CLK    CNTL          POS_E
XS_H1     VOUTA  CNTL   VOUT   SH

BTH0      D0     0      V =    u( V(VIN) -1/2)
BOFF0     VIN2   OFF0   V =    V(D0)/2
BTH1      D1     0      V =    u( V(OFF0) -1/4)
BOFF1     OFF0   OFF1   V =    V(D1)/4
BTH2      D2     0      V =    u( V(OFF1) -1/8)
BOFF2     OFF1   OFF2   V =    V(D2)/8
BTH3      D3     0      V =    u( V(OFF2) -1/16)
BOFF3     OFF2   OFF3   V =    V(D3)/16
BTH4      D4     0      V =    u( V(OFF3) -1/32)
BOFF4     OFF3   OFF4   V =    V(D4)/32
BTH5      D5     0      V =    u( V(OFF4) -1/64)
BOFF5     OFF4   OFF5   V =    V(D5)/64
BTH6      D6     0      V =    u( V(OFF5) -1/128)
BOFF6     OFF5   OFF6   V =    V(D6)/128
BTH7      D7     0      V =    u( V(OFF6) -1/256)
BOFF7     OFF6   OFF7   V =    V(D7)/245

BDAC      VOUTA  0      V =    V(D0)/2+V(D1)/4+V(D2)/8+V(D3)/16+V(D4)/32+V(D5)/64+V(D6)/128+V(D7)/245 -V(V1)
```

```
.control

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] = 2m*(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



*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      .05m    1      0      .05m
set       pensize = 2
plot       vouta vout vin xlimit 1ms 1

plot      vin -vout xlimit 1ms 1


echo                "===================FFT_and_Plot====================================="
linearize
let                 FFT_BandWidth_Hz =  1k
let                 FFT_resolution_Hz = 1
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(vout)
plot                mag (vout)   loglog
*plot               mag (vout)  ylog xlimit 400 600
echo                "plot               fft eye_chart"
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       100Meg
 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_12.41PM

dsauersanjose@aol.com
Don Sauer