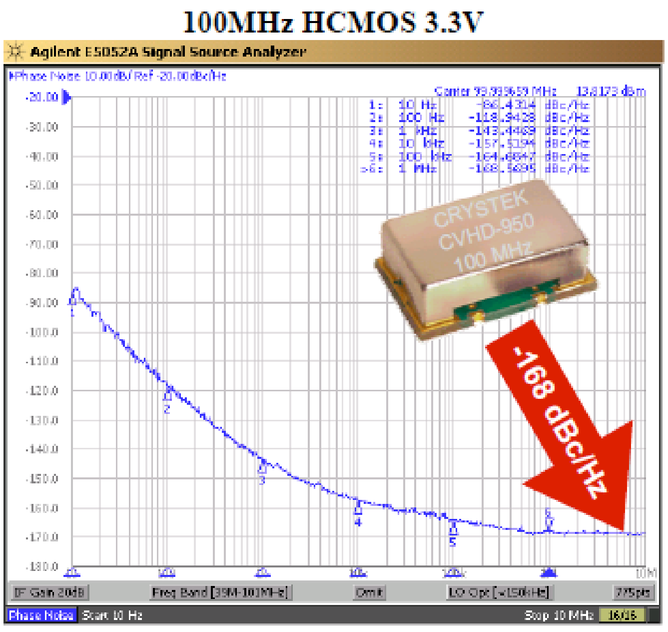
=====THE COST OF LOW PHASE NOISE===========

Now compare a high price crystal oscillator from a low price one. Crystek's oscillator family provide two such curves with some explanations.



Crystek's oscillator family CCHD-950 (clock) and CVHD-950 (VCXO) were designed as cost-effective, clean, low jitter clocks and VCXOs. This family of oscillators uses discrete components to achieve **"sub-picosecond" jitter**. Figures 6 and 7 are actual SSB phase noise plots of a commodity clock and the CCHD-950 at 100 MHz. Note that when comparing jitter specs from different oscillators, it is not sufficient to simply look at the quoted jitter of 1 ps rms, max. (12 kHz to 20 MHz). Both oscillators in Figures 6 and 7 will meet this spec, but clearly the CCHD-950 is a superior oscillator in terms of phase noise and wideband jitter.

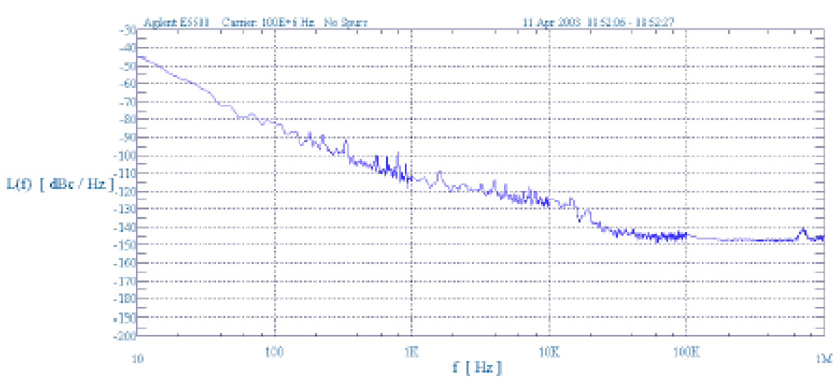


Figure 6. SSB phase noise plot of a commodity clock.

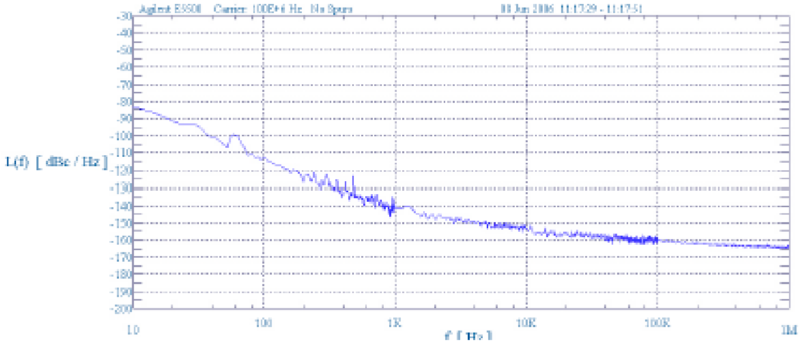
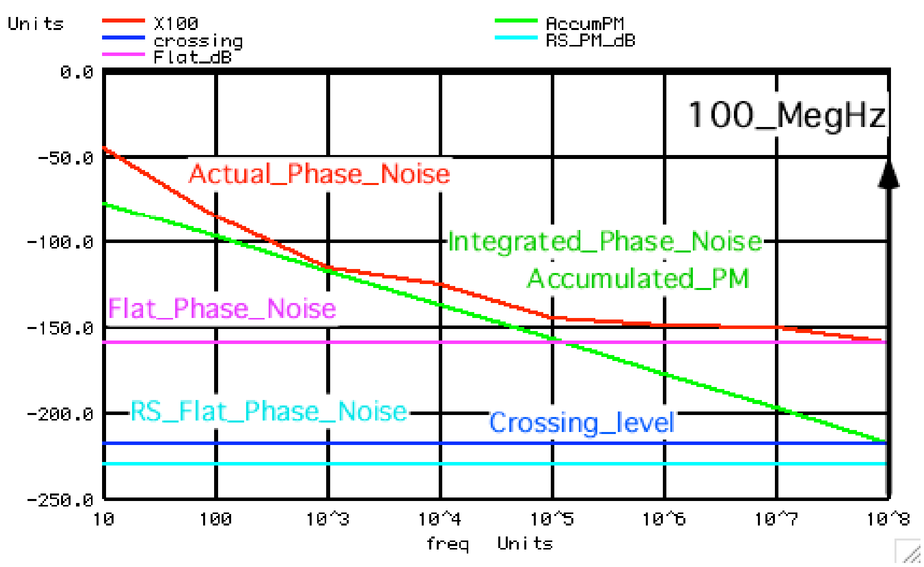


Figure 7. SSB phase noise plot of a true ultra-low phase noise oscillator (model: Crystek CCHD-950).

Achieving ultralow phase noise

A **commodity oscillator** is nothing more than an ASIC and a quartz crystal blank. **In most cases, it does not even have an internal bypass capacitor**. The crystal blank is an AT-cut strip with **Q of about 25 K to 45 K**. This low Q limits the close-in phase noise. The ASIC with all its transistors limits the floor noise to about -150 dBc/Hz. On the other hand, the true ultralow phase noise oscillator uses a discrete highperformance oscillator topology with a packaged crystal with a **Q greater than 70 K f**or excellent close-in phase noise. The discrete oscillator topology establishes the SNR, and hence the floor is lower than -160 dBc/Hz. Therefore, superior performance is obtained with very high Q crystals and a good discrete topology. This lower phase noise does come with a **price delta of approximately $15.** However, this a small price to pay (in most cases) considering the improvement gained.

Model the commodity oscillator..



.

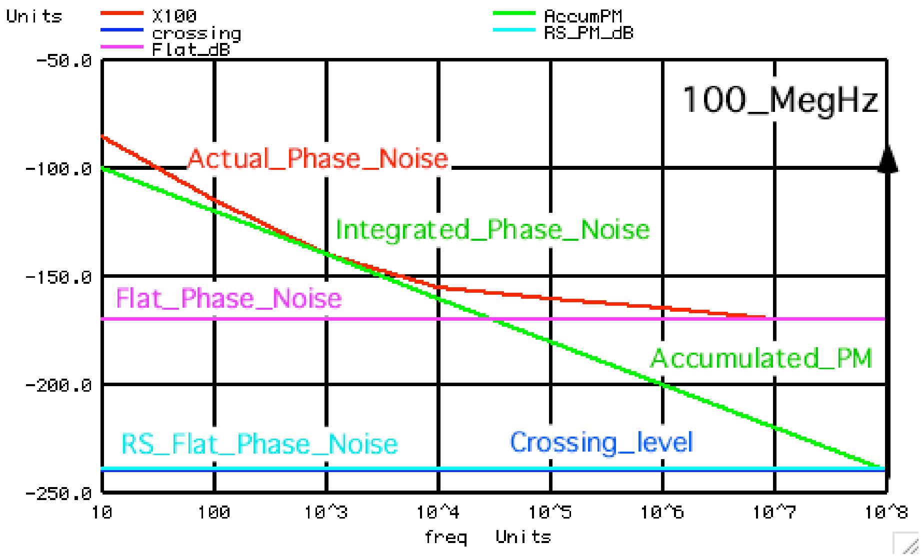
```
                                     _   _   _
                                    / \_/ \_/ \
 |‾‾‾|   /\   /\   /\    ||        |  ()  ()   |    |‾‾‾|
 |1  |__/  \ /  \ /  \___||________|           |____| 2 |
 |___|  |   \/   \/      ||                    |    |___|
        |  RS 22         CS       LS 1.2m      |
        |              0.002pF                 |   Freq  = 100Meg
        |                                      |   Z0    = 6.18Meg
        |____||________________________________|   Q     = 24K
             ||  CP 4.5pF                          CLoop = 0.001998pF
```

The integrated phase noise crossing at 100MHz is about 12dB higher than what the RS of the tuned circuit predicts it to be. This oscillator has been designed for low cost.

Circuit: Phase_Noise_100MHz_Crystal **Normal**

```
========================Create_Accumulation_Slope==============
========================Osillator_Freq_and_Period==============
OscillatorFreq_Hz =       1E+08
Oscillator_Period_s =     1E-08
========================Reference_Oscillator_Magnitude=====================
Osc_V_rms =               1
Osc_V_ppk =               2.82
Osc_db =                  0
========================Osillator_FlatNoise=====================
Osc_Flat_Noise_V_dB =     -159
Flat_Noise_V_per_Hz =     1.12202E-08
Equivalent_Noise_R  =     7868.28
One_Rad_Ref_dB      =     -db(sqrt(2)*sqrt(oscFreq))
One_Rad_Ref_dB      =     -83.0103
Jitt_floor_Rads_rms =     0.000158677
Flat_Jitter_rms_s   =     (period*jitt_Rad )/6.28
Flat_Jitter_rms_s   =     2.52671E-13
========================Crossing_At_Osc_Freq=================
Flat_Noise_PM_dB    =     where AccumPM_crosses 100Meg
Flat_Noise_PM_dB    =     -217
========================Listed_Q=======
Crystal_Q           =     24000
Rs_Ohm              =     22
========================Find_RS_FlatBand_Thermal============
Therm_Noise_RS_Hz   =     4e-9*sqrt(RS/1000)
Therm_Noise_RS_Hz   =     7.44043E-10
Therm_Noise_RS_dB   =     -182.568
========================Find_Q_Bandwidth==================
BandWidth_for_Q_Hz  =     oscFreq/Q
BandWidth_Hz        =     4166.67
========================Find_Noise_withing Bandwidth========
Rs_Noise_in_BW_rms  =     RS_noise*sqrt(BW)
Rs_Noise_in_BW_rms  =     4.80278E-08
========================Half_Noise_is_PM=======
Rs_PM_Noise_rms     =     RS_rms/sqrt(2)
Rs_PM_Noise_rms     =     3.39608E-08
Spread_out_over_Hz  =     1E+08
========================Referenced_to_0dB=======
If_Crystal_level    =     1V_rms
Expected_PM_db      =     -229.38
Flat_Noise_PM_dB    =     where AccumPM_crosses 100Meg
Flat_Noise_PM_dB    =     -217
========================Power_at_1Vrms=======
Crystal_R_parallel  =     RS*(1+CP/CS)^2
Crystal_R_parallel  =     6.13E+06
Crystal_current     =     1.63132E-07
Crystal_Power_uW    =     0.163132
========================done=======
MacSpice 40 ->
```

Now Model the oscillator that costs $15 more.




```
                                          _   _   _
                                        / \/ \/ \
|‾‾‾|  /\  /\  /\ ||     | () () |   |‾‾‾|
| 1 |_/  \/  \/  \_||_____|       |___| 2 |
|___| |      RS 11       CS      LS 1.2m    | |___|
      |              0.002pF                 |     Freq  = 100Meg
      |                                      |     Z0    = 6.18Meg
      |____||_______________________________|     Q     = 70K
           ||  CP 4.5pF                            CLoop = 0.001998pF
```

The predicted crossing is almost identical to the actual 100MHz. This says the phase noise has been pushed to the limits defined by the crystal by itself.

And money will be spend on the flat noise as well. Doing so can reduce to actual rms jitter which is dominating the overall jitter.

Circuit: Phase_Noise_100MHz_Crystal **Ultra**

```
========================Create_Accumulation_Slope=============
========================Osillator_Freq_and_Period=============
OscillatorFreq_Hz =      1E+08
Oscillator_Period_s =    1E-08
========================Reference_Oscillator_Magnitude=====================
Osc_V_rms =              1
Osc_V_ppk =              2.82
Osc_db =                 0
========================Osillator_FlatNoise=====================
Osc_Flat_Noise_V_dB =    -170
Flat_Noise_V_per_Hz =    3.16228E-09
Equivalent_Noise_R  =    625
One_Rad_Ref_dB      =    -db(sqrt(2)*sqrt(oscFreq))
One_Rad_Ref_dB      =    -83.0103
Jitt_floor_Rads_rms =    4.47214E-05
Flat_Jitter_rms_s   =    (period*jitt_Rad )/6.28
Flat_Jitter_rms_s   =    7.12124E-14
========================Crossing_At_Osc_Freq================
Flat_Noise_PM_dB    =    where AccumPM_crosses 100Meg
Flat_Noise_PM_dB    =    -240
========================Listed_Q=======
Crystal_Q           =    70000
```

```
Rs_Ohm                 =   11
========================Find_RS_FlatBand_Thermal============
Therm_Noise_RS_Hz    =   4e-9*sqrt(RS/1000)
Therm_Noise_RS_Hz    =   4.19524E-10
Therm_Noise_RS_dB    =   -187.545
========================Find_Q_Bandwidth===================
BandWidth_for_Q_Hz   =   oscFreq/Q
BandWidth_Hz         =   1428.57
========================Find_Noise_within Bandwidth========
Rs_Noise_in_BW_rms   =   RS_noise*sqrt(BW)
Rs_Noise_in_BW_rms   =   1.58565E-08
========================Half_Noise_is_PM=======
Rs_PM_Noise_rms      =   RS_rms/sqrt(2)
Rs_PM_Noise_rms      =   1.12122E-08
Spread_out_over_Hz   =   1E+08
========================Referenced_to_0dB=======
If_Crystal_level     =   1V_rms
Expected_PM_db       =   -239.006
Flat_Noise_PM_dB     =   where AccumPM_crosses 100Meg
Flat_Noise_PM_dB     =   -240
========================Power_at_1Vrms=======
Crystal_R_parallel   =   RS*(1+CP/CS)^2
Crystal_R_parallel   =   6.13E+06
Crystal_current      =   1.63132E-07
Crystal_Power_uW     =   0.163132
========================done=======
```

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

Phase_Noise_100MHz_Crystal Normal

```
*=========Create_Signal_No_Reason==================
*V_SIN#    NODE_P NODE_N DC     VALUE  SIN(   V_DC   AC_MAG FREQ   DELAY  FDamp)
VIN        VP     0      DC     0      SIN(   0      1      1                  )

.control
set      pensize = 2
unlet    stanDev_val
unlet    Out_percent
unlet    X100
unlet    freq
unlet    intnoise
let      X100             = vector(8)
let      freq            = vector(8)
let      intnoise        = vector(8)

let      offsett = -57

let      freq[0] = 10
let      X100[0] = -45
let      freq[1] = 100
let      X100[1] = -85
let      freq[2] = 1k
let      X100[2] = -115
let      freq[3] = 10k
let      X100[3] = -125
let      freq[4] = 100k
let      X100[4] = -145
let      freq[5] = 1Meg
let      X100[5] = -149
let      freq[6] = 10Meg
let      X100[6] = -150
let      freq[7] = 100Meg
let      X100[7] = -159

echo             "=======================Create_Timing_Slope============="
let      index = 0
repeat           8
let              intnoise[index] = -20*log(freq[index]) + offsett
let index =      index +1
end
```

```
echo             "=======================Osillator_Freq_and_Period============="
let oscFreq =    100meg
echo             "OscillatorFreq_Hz =     $&oscFreq"
let period  =    1/oscFreq
echo             "Oscillator_Period_s =   $&period"
echo             "=======================Reference_Oscillator_Magnitude==================
=="
let oscVppk =    2.82
let oscVrms =    1
let osc_db  =    db(oscVrms)
echo             "Osc_V_rms =             $&oscVrms"
echo             "Osc_V_ppk =             $&oscVppk"
echo             "Osc_db =                $&osc_db"
echo             "=======================Osillator_FlatNoise===================="
let Flat_dB =    X100[7]
echo             "Osc_Flat_Noise_V_dB =   $&Flat_dB"
let Flat_V  =    1/exp(ln(10)*abs(Flat_dB/20))
echo             "Flat_Noise_V_per_Hz =   $&Flat_V"
let Eq_R =        (Flat_V/4n)*(Flat_V/4n)*1k
echo             "Equivalent_Noise_R  =   $&Eq_R"

let Rad_REF_db =  -db(sqrt(2)*sqrt(oscFreq))
echo             "One_Rad_Ref_dB      =   -db(sqrt(2)*sqrt(oscFreq))"
echo             "One_Rad_Ref_dB      =   $&Rad_REF_db"

let jitt_Rad  =   1/exp(ln(10)*abs((Rad_REF_db-Flat_dB)/20))
echo             "Jitt_floor_Rads_rms =   $&jitt_Rad "

let jit_s    =   (period* jitt_Rad )/6.28
echo             "Flat_Jitter_rms_s   =   (period*jitt_Rad )/6.28"
echo             "Flat_Jitter_rms_s   =   $&jit_s"
echo             "=======================Timing_Tolerance_At_Osc_Freq==========="
let expectN =    -20*log(oscFreq) + offsett
echo             "Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "=======================Listed_Q======="
let Q =          24K
echo             "Crystal_Q           =   $&Q"
let RS =         22
echo             "Rs_Ohm              =   $&RS"
echo             "=======================Find_RS_FlatBand_Thermal==========="
let RS_noise =   4e-9*sqrt(34.6/1000)
echo             "Therm_Noise_RS_Hz   =   4e-9*sqrt(RS/1000)"
echo             "Therm_Noise_RS_Hz   =   $&RS_noise"
let RS_nois_dB =  db(RS_noise)
echo             "Therm_Noise_RS_dB   =   $&RS_nois_dB"
echo             "=======================Find_Q_Bandwidth=================="
let BW =          oscFreq/Q
echo             "BandWidth_for_Q_Hz  =   oscFreq/Q"
echo             "BandWidth_Hz        =   $&BW"
echo             "=======================Find_Noise_withing Bandwidth======="
let RS_rms =      RS_noise*sqrt(BW)
echo             "Rs_Noise_in_BW_rms  =   RS_noise*sqrt(BW)"
echo             "Rs_Noise_in_BW_rms  =   $&RS_rms"
echo             "=======================Half_Noise_is_PM======="
let RS_PM_rms =   RS_rms/sqrt(2)
echo             "Rs_PM_Noise_rms     =   RS_rms/sqrt(2)"
echo             "Rs_PM_Noise_rms     =   $&RS_PM_rms"
echo             "Spread_out_over_Hz  =   $&oscFreq"
echo             "=======================Half_Noise_is_PM======="
let TimeToler =   RS_PM_rms/sqrt(oscFreq)
let RS_PM_dB =    db(TimeToler)
echo             "If_Crystal_level    =   1V_rms"
echo             "Expected_PM_db      =   $&RS_PM_dB"
echo             "Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "=======================Power_at_1Vrms======="
let RP  =         6130K
echo             "Crystal_R_parallel  =   RS*(1+CP/CS)^2"
echo             "Crystal_R_parallel  =   $&RP"
```

```
let Icryst  =     1/RP
echo              "Crystal_current     =   $&Icryst"
let Pcryst  =     Icryst/1u
echo              "Crystal_Power_uW    =   $&Pcryst"

echo              "========================done======="

let AccumPM =     intnoise
let crossing =    expectN
plot              X100 AccumPM crossing RS_PM_dB Flat_dB vs freq xlog
plot              X100 - AccumPM  vs freq xlog

.endc
.end
```

Phase_Noise_100MHz_Crystal Ultra

```
*=========Create_Signal_No_Reason==================
*V_SIN#    NODE_P NODE_N DC      VALUE   SIN(    V_DC    AC_MAG FREQ    DELAY   FDamp)
VIN        VP     0      DC      0       SIN(    0       1      1                     )

.control
set      pensize = 2
unlet    stanDev_val
unlet    Out_percent
unlet    X100
unlet    freq
unlet    intnoise
let      X100             = vector(8)
let      freq            = vector(8)
let      intnoise        = vector(8)

let      offsett = -80

let      freq[0] = 10
let      X100[0] = -85
let      freq[1] = 100
let      X100[1] = -115
let      freq[2] = 1k
let      X100[2] = -140
let      freq[3] = 10k
let      X100[3] = -155
let      freq[4] = 100k
let      X100[4] = -160
let      freq[5] = 1Meg
let      X100[5] = -165
let      freq[6] = 10Meg
let      X100[6] = -170
let      freq[7] = 100Meg
let      X100[7] = -170

echo              "========================Create_Timing_Slope=============="
let      index = 0
repeat            8
let               intnoise[index] = -20*log(freq[index]) + offsett
let index =       index +1
end

echo              "========================Osillator_Freq_and_Period=============="
let oscFreq =     100meg
echo              "OscillatorFreq_Hz =     $&oscFreq"
let period  =     1/oscFreq
echo              "Oscillator_Period_s =   $&period"
echo              "========================Reference_Oscillator_Magnitude===================
=="
let oscVppk =     2.82
let oscVrms =     1
let osc_db  =     db(oscVrms)
echo              "Osc_V_rms =             $&oscVrms"
echo              "Osc_V_ppk =             $&oscVppk"
echo              "Osc_db =                $&osc_db"
```

```
echo             "=======================Osillator_FlatNoise===================="
let Flat_dB =    X100[7]
echo             "Osc_Flat_Noise_V_dB =   $&Flat_dB"
let Flat_V  =    1/exp(ln(10)*abs(Flat_dB/20))
echo             "Flat_Noise_V_per_Hz =   $&Flat_V"
let Eq_R =        (Flat_V/4n)*(Flat_V/4n)*1k
echo             "Equivalent_Noise_R  =   $&Eq_R"

let Rad_REF_db =  -db(sqrt(2)*sqrt(oscFreq))
echo             "One_Rad_Ref_dB      =   -db(sqrt(2)*sqrt(oscFreq))"
echo             "One_Rad_Ref_dB      =   $&Rad_REF_db"

let jitt_Rad  =   1/exp(ln(10)*abs((Rad_REF_db-Flat_dB)/20))
echo             "Jitt_floor_Rads_rms =   $&jitt_Rad "

let jit_s    =   (period* jitt_Rad )/6.28
echo             "Flat_Jitter_rms_s   =   (period*jitt_Rad )/6.28"
echo             "Flat_Jitter_rms_s   =   $&jit_s"
echo             "=======================Timing_Tolerance_At_Osc_Freq============"
let expectN =    -20*log(oscFreq) + offsett
echo             "Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "=======================Listed_Q======="
let Q =          70K
echo             "Crystal_Q           =   $&Q"
let RS =          11
echo             "Rs_Ohm              =   $&RS"
echo             "=======================Find_RS_FlatBand_Thermal============"
let RS_noise =   4e-9*sqrt(RS/1000)
echo             "Therm_Noise_RS_Hz   =   4e-9*sqrt(RS/1000)"
echo             "Therm_Noise_RS_Hz   =   $&RS_noise"
let RS_nois_dB =  db(RS_noise)
echo             "Therm_Noise_RS_dB   =   $&RS_nois_dB"
echo             "=======================Find_Q_Bandwidth==================="
let BW =          oscFreq/Q
echo             "BandWidth_for_Q_Hz  =   oscFreq/Q"
echo             "BandWidth_Hz        =   $&BW"
echo             "=======================Find_Noise_withing Bandwidth========"
let RS_rms =      RS_noise*sqrt(BW)
echo             "Rs_Noise_in_BW_rms  =   RS_noise*sqrt(BW)"
echo             "Rs_Noise_in_BW_rms  =   $&RS_rms"
echo             "=======================Half_Noise_is_PM======="
let RS_PM_rms =   RS_rms/sqrt(2)
echo             "Rs_PM_Noise_rms     =   RS_rms/sqrt(2)"
echo             "Rs_PM_Noise_rms     =   $&RS_PM_rms"
echo             "Spread_out_over_Hz  =   $&oscFreq"
echo             "=======================Half_Noise_is_PM======="
let TimeToler =   RS_PM_rms/sqrt(oscFreq)
let RS_PM_dB =    db(TimeToler)
echo             "If_Crystal_level    =   1V_rms"
echo             "Expected_PM_db      =   $&RS_PM_dB"
echo             "Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "=======================Power_at_1Vrms======="
let RP  =         6130K
echo             "Crystal_R_parallel  =   RS*(1+CP/CS)^2"
echo             "Crystal_R_parallel  =   $&RP"
let Icryst  =     1/RP
echo             "Crystal_current     =   $&Icryst"
let Pcryst  =     Icryst/1u
echo             "Crystal_Power_uW    =   $&Pcryst"

echo             "=======================done======="

let AccumPM =    intnoise
let crossing =   expectN
plot              X100 AccumPM crossing RS_PM_dB Flat_dB vs freq xlog
plot              X100 - AccumPM  vs freq xlog

.endc
```

.end

4.18.11_1.35PM
dsauersanjose@aol.com
Don Sauer