=====SIMPLE PHASE NOISE PREDICTION FOR LC OSCILLATORS===========

Here is a popular model for phase noise of an Oscillator.
It does not claim that this model is predictive.

Leeson's model

- For the LC oscillator

$$\mathcal{L}(\Delta\omega) = 10log\left[\frac{2kT}{P_{sig}}\left(\frac{\omega_o}{2Q\Delta\omega}\right)^2\right]$$

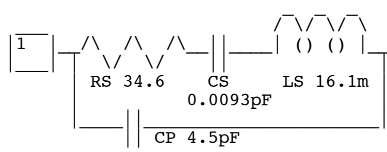
- Leeson's model includes flicker noise

$$\mathcal{L}(\Delta\omega) = 10\log\left[\frac{2FkT}{P_{sig}}\left\{1+\left(\frac{\omega_o}{2Q\Delta\omega}\right)^2\right\}\left(1+\frac{\Delta\omega_c}{(\Delta\omega)^3}\right)\right]$$

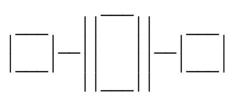
- F is an "effective noise figure" for the oscillator
- It is not a predictive model

Crystals are modeled as linear tuned circuits.
It is not obvious that phase noise can not be derived from the spice models.
There may be a simple way to predict phase noise using a graph and a calculator.

The following are some crystal tables and equations off the web.
The values in light blue are extrapolated.




```
Freq  = 13Meg
Z0    = 8.13Meg
Q     = 38K
CLoop = 0.009281pF
```



Frequency	RS	LS	CS	CP	CR*CP/(CR+CP)	CP/CS	RP	Q	Z0
MHz	Ohms	mH	pF	pF	pf		Ohms		
2.00	200	520	0.012	4	0.0119641	333	22.3Meg	33k	6582k
5.00	50	115	0.010	3	0.0099668	300	4.53Meg	67k	3396k
13.00	34.6	16.1	0.0093	4.5	0.0092808	483.87	8.13Meg	38K	1315k
15.00	30	12.5	0.009	5	0.0089838	555.55	9.29Meg	39K	1178k
30.00	20	4.7	0.006	3	0.005988	500	5.02Meg	44k	885K
100.00	11	1.2	0.002	1.5	0.001998	750	6.18Meg	70k	775K

```
Fseries    =    1/(2*PI*sqrt(CS*LS))
Cloop      =    CR*CP/(CR+CP)
Fparallel  =    1/(2*PI*sqrt(Cloop *LS))
RP         =    RS*(1+CP/CS)^2
Q          =    2*PI*Fseries*LS/RS
```

Look at the Phase Noise Plot of a real 13MHz Crystal Oscillator.

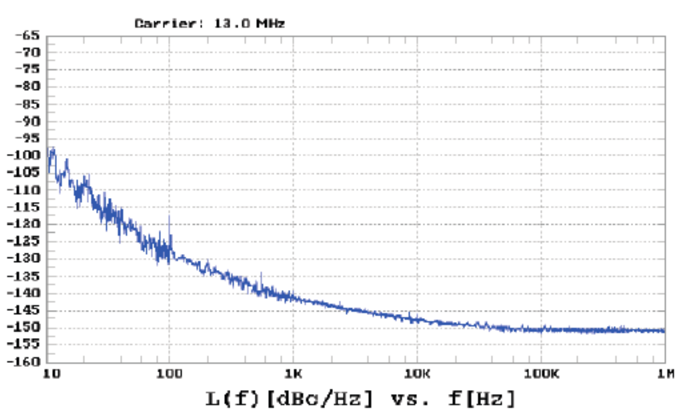
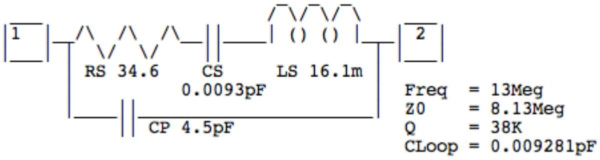
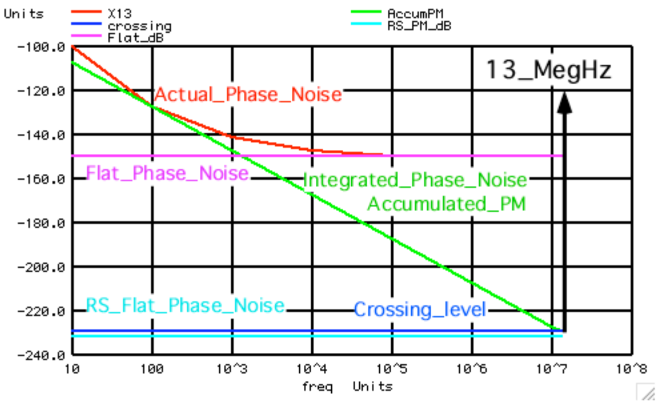


Fig.8 is a Phase Noise plot of a real 13.0MHz Crystal Oscillator.

Here is the spice model for just the crystal.



Add a horizontal Phase Noise line which describes the flat band phase noise.
Now add a vertical line at 13MegHz to show the oscillation frequency.



Now added a 20db/dec slope curve such that it is tangent to the Actual Phase Noise curve.
That slope will be called the Integrated or Accumulated Phase Noise level.
In an oscillator, the phase noise of each cycle gets accumulated over time.
It crosses 13Mhz at -229 dB which will be called the crossing level.

When looking at the Actual Phase Noise curve at 10Hz,
one cannot detect something happening at 10Hz with out taking 100ms worth of data.

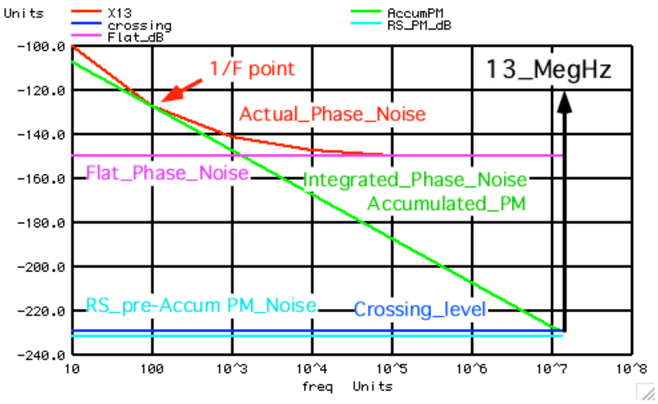
In other words, 1.3 million clock cycles will be needed to measure something at 10Hz.
The pre-accumulated phase noise of each cycle will accumulate over that 100msec period.
It turns out rms power increases by the square root of the number of clock cycles.
So think of Phase noise in terms of the accumulation that is taking place.

Filter the thermal noise of RS by the bandwidth of the Crystal.
Assume the crystal is running at 1Vrms which is 0dB.
Only half of this filtered RMS voltage is going into Phase Modulation.
The resulting rms voltage is then being sampled at 13MHz Nyquist.
The resulting rms voltage will be spread uniformly over Nyquist at 13MHz.
This produces a pre-accumulated PM level at -231 dB.
Notice it is almost the same as the crossing level.

When a pre-accumulated PM noise level get accumulated,
it will equal the accumulation PM level when looking over a period of one clock cycle.
No accumulations will be taking place.
This is the jitter one sees when viewing and synching one cycle on a scope.

So the non-accumulated phase noise due to the RS thermal noise
should match the accumulated phase Noise when crossing 13MHz.

This example is assuming the crystal is seeing 1Vrms across it.
This oscillator runs at 3.3V.
That comes close to a 2.82Vppk swing across the crystal.



So given the model of the tuned circuit and the voltage across the tuned circuit,
an integrated or accumulated line having a 20dB per decade slope
can be drawn from where the clock frequency crosses
the filtered and sampled pre-accumulated thermal noise level of the tuned circuit.

The drawn integrated or accumulated line describes the limitations of the crystal.
The flat phase noise level will be present due to things outside the tuned circuit.
A 1KOhm resistor has 4nV per root hertz noise.
Transistors produce almost the same noise for a given impedance.
It takes some supply current to get transistors to operate at low impedances.
For instance, a bipolar transistor needs to draw 1mA to have the noise of 13 Ohms.

Above the 1/f point, accumulated pm noise and supply current might defined Phase Noise.
Bipolar transistor have much lower 1/f noise than CMOS transistors.
The 1/f point can be lowered further using some larger transistors.

```
========================Create_Accumulation_Slope==============
Oscillator
OscillatorFreq_Hz    =   1.3E+07
Oscillator_Period_s  =   7.69231E-08
========================Reference_Oscillator_Magnitude=========
Osc_V_rms            =   1
Osc_V_ppk            =   2.82
Osc_db               =   0
========================Osillator_FlatNoise====================
Osc_Flat_Noise_V_dB  =   -150
Flat_Noise_V_per_Hz  =   3.16228E-08
Equivalent_Noise_R   =   62500
One_Rad_Ref_dB       =   -db(sqrt(2)*sqrt(oscFreq))
One_Rad_Ref_dB       =   -74.1497
Jitt_floor_Rads_rms  =   0.000161245
Flat_Jitter_rms_s    =   (period*jitt_Rad )/6.28
Flat_Jitter_rms_s    =   1.97508E-12
========================Crossing_At_Osc_Freq==================
Flat_Noise_PM_dB     =   where AccumPM crosses 13Meg
Flat_Noise_PM_dB     =   -229.279
========================Listed_Q=======
Crystal_Q            =   38000
Rs_Ohm               =   34.6
========================Find_RS_FlatBand_Thermal============
Therm_Noise_RS_Hz    =   4e-9*sqrt(34.6/1000)
Therm_Noise_RS_Hz    =   7.44043E-10
Therm_Noise_RS_dB    =   -182.568
========================Find_Q_Bandwidth===================
BandWidth_for_Q_Hz   =   oscFreq/Q
BandWidth_Hz         =   342.105
========================Find_Noise_withing Bandwidth========
Rs_Noise_in_BW_rms   =   RS_noise*sqrt(BW)
Rs_Noise_in_BW_rms   =   1.37619E-08
========================Half_Noise_is_PM=======
Rs_PM_Noise_rms      =   RS_rms/sqrt(2)
Rs_PM_Noise_rms      =   9.73112E-09
Spread_out_over_Hz   =   1.3E+07
========================Referenced_to_0dB=======
If_Crystal_level     =   1V_rms
Expected_PM_db       =   -231.376
Flat_Noise_PM_dB     =   where AccumPM_crosses 13Meg
Flat_Noise_PM_dB     =   -229.279
========================Power_at_1Vrms=======
Crystal_R_parallel   =   RS*(1+CP/CS)^2
Crystal_R_parallel   =   8.13E+06
Crystal_current      =   1.23001E-07
Crystal_Power_uW     =   0.123001
========================done=======
```

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

Phase_Noise_13MHz_Crystal

```
*=========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    X13
unlet    freq
unlet    intnoise
let      X13             = vector(8)
let      freq            = vector(8)
let      intnoise        = vector(8)

let      offsett = -87

let      freq[0] = 10
let      X13[0] = -100
let      freq[1] = 100
let      X13[1] = -127
let      freq[2] = 1k
let      X13[2] = -141
let      freq[3] = 10k
let      X13[3] = -147
let      freq[4] = 100k
let      X13[4] = -150
let      freq[5] = 1Meg
let      X13[5] = -150
let      freq[6] = 10Meg
let      X13[6] = -150
let      freq[7] = 13Meg
let      X13[7] = -150

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 =    13meg
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 =    X13[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 13Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "========================Listed_Q======="
let Q =          38K
echo             "Crystal_Q           =   $&Q"
let RS =         34.6
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(34.6/1000)"
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             "========================Referenced_to_0dB======="
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 13Meg "
```

```
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "========================Power_at_1Vrms======="
let RP  =         813OK
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              X13 AccumPM crossing RS_PM_dB Flat_dB vs freq xlog
plot              X13 - AccumPM  vs freq xlog

.endc
.end
```

4.18.11_1.36PM

dsauersanjose@aol.com
Don Sauer