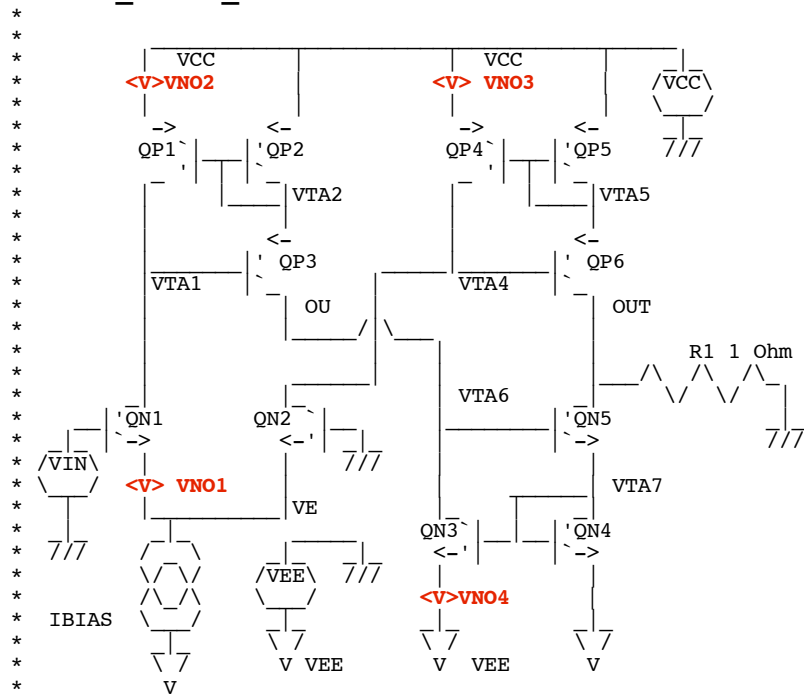


LM3080_Audio_OffsetNoise



```
.OPTIONS GMIN=1e-12 METHOD=trap srcsteps = 1 gminsteps = 1
.OPTIONS ITL1=400 ITL2=100 ITL6=100
.OPTIONS RELTOL=.001 ABSTOL=1p VNTOL=1u
```

=====					
VCC	VCC	0	DC	5	
VEE	0	VEE	DC	5	
VIN	VIN	0	DC	0	
VNO1	VNO1	VE	DC	0	
VNO2	VNO2	VCC	DC	0	
VNO3	VNO3	VCC	DC	0	
VNO4	VNO4	VEE	DC	0	
IBIAS	VE	0	DC	1u	
QN1	VTA1	VIN	VNO1	NPN1	2
QN2	VTA4	0	VE	NPN1	2
QP1	VTA1	VTA2	VNO2	PNP1	1
QP2	VTA2	VTA2	VCC	PNP1	1
QP3	VTA6	VTA1	VTA2	PNP1	1
QP4	VTA4	VTA5	VNO3	PNP1	1
QP5	VTA5	VTA5	VCC	PNP1	1
QP6	OUT	VTA4	VTA5	PNP1	1
QN3	VTA6	VTA7	VNO4	NPN1	1
QN4	VTA7	VTA7	VEE	NPN1	1
QN5	OUT	VTA6	VTA7	NPN1	1
R1	OUT	0	1		

```
.control
setplot new
set thisData = data
let $thisData = 0 * vector(18)
set thisData2 = data2
let $thisData2 = 0 * vector(18)

let n=0
repeat 17
let unknown.{$thisData2}[n] = n
let n=n+1
endrepeat
```

```
=====Geometry Sensitivity=====
* Add a voltage source in series with one transistor for each matched pair
* measure the effect of adding a standart error (1mV or 1%)
* Store all gain and offset results for each transistor
* Calibrate each transistor to its area and processing expectations
```

```
*#0=====Define 1X_NPN_and_PNPL_and_RB_Offset=====
let npnoff = .6m
let pnpoff = 1.1m
let rboff = .005
```

```

let      unknown.{\$thisData}[0] = npnoff
let      unknown.{\$thisData}[1] = pnpoff
let      unknown.{\$thisData}[2] = rbofff
let      npnbf = @NPN1[bf]
let      pnpbf = @PNP1[bf]
let      npnrb = @NPN1[rb]
let      pnprb = @PNP1[rb]

echo      "Process Expectations  "
echo      "NPN_offset   =  \$&nnpoff  "
echo      "NPN_beta     =  \$&nnpbf   "
echo      "NPN_rbase    =  \$&nnpnr   "
echo      "PNPL_offset  =  \$&pnpoff  "
echo      "PNPL_beta   =  \$&pnpbf   "
echo      "PNPL_rbase  =  \$&pnprb   "
echo      "Rbase_offset =  \$&rbofff  "
echo      ""

*#0=====Measure_With_No_Errors=====
op
let      unknown.{\$thisData}[3] = out
let      ibb      = @ibias[dc]
let      icc      = -1*vcc#branch
echo      "Ibias      =  \$&ibb   "
echo      "Isupply    =  \$&icc   "
echo      ""

*=====Measure_Offset_Gain_Each_Transistor=====
* Add 1mV offset one at a time to each transistor
* Measure change at output
* Store gain and offset results for each transistor
* measure transistor area
* calculate each transistors offset from area and npnoff, pnpoff, and rbofff

*#1=====Measure_OTA_gm_At_1mV=====
alter    vin      dc = 1m
op
let      gm      = abs(out- unknown.{\$thisData}[3]) /1m
let      unknown.{\$thisData}[4] = gm
alter    vin      dc = 0

*#2=====Measure_VN01_effect=====
alter    vno1     dc = 1m
op
let      procOff = unknown.{\$thisData}[0]
let      Tarea  = @qn1[area]
let      Ioutref = unknown.{\$thisData}[3]
let      gm      = unknown.{\$thisData}[4]
let      vingain = 1000*(out - Ioutref)/gm
let      pairOff = (1/Tarea)^.5*procOff
let      vinoff  = pairOff*vingain
let      unknown.{\$thisData}[5] = vingain
let      unknown.{\$thisData}[6] = vinoff
echo      "Pair      ProcOff Area Pair_Off Gain2Vin Vin_Offset "
echo      "QN1/QN2  \$&procOff  \$&Tarea  \$&pairOff  \$&vingain  \$&vinoff"

*#3=====Measure_VN02_effect=====
alter    vno1     dc = 0
alter    vno2     dc = 1m
op
let      procOff = unknown.{\$thisData}[1]
let      Tarea  = @qp1[area]
let      Ioutref = unknown.{\$thisData}[3]
let      gm      = unknown.{\$thisData}[4]
let      vingain = 1000*(out - Ioutref)/gm
let      pairOff = (1/Tarea)^.5*procOff
let      vinoff  = pairOff*vingain
let      unknown.{\$thisData}[7] = vingain
let      unknown.{\$thisData}[8] = vinoff
echo      "QP1/QP2  \$&procOff  \$&Tarea  \$&pairOff  \$&vingain  \$&vinoff"

*#4=====Measure_VN03_effect=====
alter    vno2     dc = 0
alter    vno3     dc = 1m
op
let      procOff = unknown.{\$thisData}[1]
let      Tarea  = @qp4[area]
let      Ioutref = unknown.{\$thisData}[3]
let      gm      = unknown.{\$thisData}[4]
let      vingain = 1000*(out - Ioutref)/gm
let      pairOff = (1/Tarea)^.5*procOff
let      vinoff  = pairOff*vingain
let      unknown.{\$thisData}[9] = vingain

```

```

let      unknown.{\$thisData}[10] = vinoff
echo     "QP4/QP5  \$&procOff  \$&Tarea  \$&pairOff  \$&vingain  \$&vinoff"

*#4=====Measure_VNO4_effect=====
alter   vno3  dc = 0
alter   vno4  dc = 1m
op
let      procOff = unknown.{\$thisData}[0]
let      Tarea  = @qn3[area]
let      Ioutref = unknown.{\$thisData}[3]
let      gm      = unknown.{\$thisData}[4]
let      vingain = 1000*(out - Ioutref)/gm
let      pairOff = (1/Tarea)^.5*procOff
let      vinoff  = pairOff*vingain
let      unknown.{\$thisData}[11] = vingain
let      unknown.{\$thisData}[12] = vinoff
echo     "QN3/QN4  \$&procOff  \$&Tarea  \$&pairOff  \$&vingain  \$&vinoff"

*#6=====Add_Up_total_offset=====
let      off1    = unknown.{\$thisData}[6]
let      off2    = unknown.{\$thisData}[8]
let      off3    = unknown.{\$thisData}[10]
let      off4    = unknown.{\$thisData}[12]
let      Vin_total = ( off1^2 + off2^2 + off3^2 + off4^2 )^.5

echo     " "
echo     "Total referred to input offset  \$&Vin_total"
echo     " "

*#8=====Measure_Maximum_Iout=====
alter   vno4  dc = 0
alter   vin   dc = -130m
op
let      unknown.{\$thisData}[12] = out
alter   vin   dc = 0
op

*=====Calculating_Noise_Each_Transistor=====
* Find Area and Emitter current for each transistor (Qn)
* Noise Resistance = (kt/q)/(2*Ie) + Rb/area
* Input_noise_Qn = 4.07nV/rHz*sqrt(Noise Resistance/1000)
* output_noise_Qn = output_noiseQn*Gain_Qn

*#7=====Calc_VNO1_effect_On_Noise=====
let      icc     = @qn1[ic]
let      Tarea  = @qn1[area]
let      rbb    = @NPN1[rb]/Tarea
let      NoiseR = rbb + 13m/icc
let      NoiseP = 4.07n*(2*NoiseR/1000)^.5
let      Gain2in = unknown.{\$thisData}[5]
let      VinNois = abs(NoiseP*Gain2in)
let      gm      = unknown.{\$thisData}[4]
let      IoutNo  = VinNois*gm
let      unknown.{\$thisData}[13] = VinNois
echo     "Pair  Ic      NoiseRes  Pair_Noise  Gain2Vin  VnoiseIn_rHz  OutNoise_rHz"
echo     "QN1/QN2  \$&icc  \$&NoiseR  \$&NoiseP  \$&Gain2in  \$&VinNois  \$&IoutNo"

*#8=====Calc_VNO2_effect_On_Noise=====
let      icc     = @qp1[ic]
let      Tarea  = @qp1[area]
let      rbb    = @PNP1[rb]/Tarea
let      NoiseR = rbb + 13m/icc
let      NoiseP = 4.07e-9*(2*NoiseR/1000)^.5
let      Gain2in = unknown.{\$thisData}[7]
let      VinNois = abs(NoiseP*Gain2in)
let      gm      = unknown.{\$thisData}[4]
let      IoutNo  = VinNois*gm
let      unknown.{\$thisData}[14] = VinNois
echo     "QP1/QP2  \$&icc  \$&NoiseR  \$&NoiseP  \$&Gain2in  \$&VinNois  \$&IoutNo"

*#9=====Calc_VNO3_effect_On_Noise=====
let      icc     = @qp4[ic]
let      Tarea  = @qp4[area]
let      rbb    = @PNP1[rb]/Tarea
let      NoiseR = rbb + 13m/icc
let      NoiseP = 4.07e-9*(2*NoiseR/1000)^.5
let      Gain2in = unknown.{\$thisData}[9]
let      VinNois = abs(NoiseP*Gain2in)
let      gm      = unknown.{\$thisData}[4]
let      IoutNo  = VinNois*gm
let      unknown.{\$thisData}[15] = VinNois
echo     "QP4/QP5  \$&icc  \$&NoiseR  \$&NoiseP  \$&Gain2in  \$&VinNois  \$&IoutNo"

*#10=====Calc_VNO4_effect_On_Noise=====

```

```

let      icc      = @qn3[ic]
let      Tarea   = @qn3[area]
let      rbb     = @NPN1[rb]/Tarea
let      NoiseR  = rbb + 13m/icc
let      NoiseP  = 4.07e-9*(2*NoiseR/1000)^.5
let      Gain2in = unknown.{ $thisData}[9]
let      VinNois = abs(NoiseP*Gain2in)
let      gm      = unknown.{ $thisData}[4]
let      IoutNo  = VinNois*gm
let      unknown.{ $thisData}[16] = VinNois
echo     "QN3/QN4  $&icc  $&NoiseR  $&NoiseP  $&Gain2in  $&VinNois  $&IoutNo"

```

```

*#11=====Add_Up_All_Noise=====

```

```

let      off1    = unknown.{ $thisData}[13]
let      off2    = unknown.{ $thisData}[14]
let      off3    = unknown.{ $thisData}[15]
let      off4    = unknown.{ $thisData}[16]
let      Vin_T   = ( off1^2 + off2^2 + off3^2 + off4^2 )^.5
let      Out_T   = Vin_T*gm
let      ibb     = @ibias[dc]
let      ibb     = unknown.{ $thisData}[12]
let      SigNo   = 20*log(ibb/(Out_T*141)) -3

let      Rgm     = 1/unknown.{ $thisData}[4]
let      Vin_max = ibb*Rgm

```

```

echo     " "
echo     "Rgm (vin/iout)                $&Rgm Ohms"
echo     "Total referred to input noise_rthz $&Vin_T V"
echo     "Vin maxium                      $&Vin_max V"
echo     "Total output current noise_rthz  $&Out_T A"
echo     "Clipping +/- output current      $&ibb A"
echo     "Clipping 20-20KHz output S/N     $&SigNo dB"

echo     " "

```

```

*let y = unknown.{ $thisData}
*let x = unknown.{ $thisData2}
*plot y vs x

```

```

.endc

```

```

*===== models Using Near Perfect Transistors =====
.model NPN1 NPN( BF = 200 VAF = 100 rb=400 )
.model PNP1 PNP( BF = 30 VAF = 150 rb=900 )
*=====

```

```

.end

```

```

=====END_OF_SPICE=====

```

To Covert PDF to plain text click below
<http://www.fileformat.info/convert/doc/pdf2txt.htm>

Circuit: LM3080_Audio_OffsetNoise

```

Process      Expectations
NPN_offset  = 0.0006
NPN_beta    = 200
NPN_rbase   = 400
PNPL_offset = 0.0011
PNPL_beta   = 30
PNPL_rbase  = 900
Rbase_offset = 0.005

Ibias       = 1E-06
Isupply     = 1.98706E-06

```

Pair	ProcOff	Area	Pair_Off	Gain2Vin	Vin_Offset
QN1/QN2	0.0006	2	0.00042	1.00025	0.000424371
QP1/QP2	0.0011	1	0.0011	0.91927	0.0010112
QP4/QP5	0.0011	1	0.0011	-0.92239	-0.00101463
QN3/QN4	0.0006	1	0.0006	-1.00849	-0.000605095

Total referred to input offset **0.0016119**

Pair	Ic	NoiseRes	Pair Noise	Gain2Vin	VnoiseIn rtHz	OutNoise rtHz
QN1/QN2	4.97832E-07	26313.2	2.95255E-08	1.00025	2.95329E-08	5.64032E-13
QP1/QP2	4.82191E-07	27860.3	3.0381E-08	0.91927	2.79284E-08	5.33388E-13
QP4/QP5	4.81813E-07	27881.4	3.03925E-08	-0.92239	2.80338E-08	5.35401E-13
QN3/QN4	4.94213E-07	26704.5	2.97441E-08	-0.92239	2.74357E-08	5.23978E-13

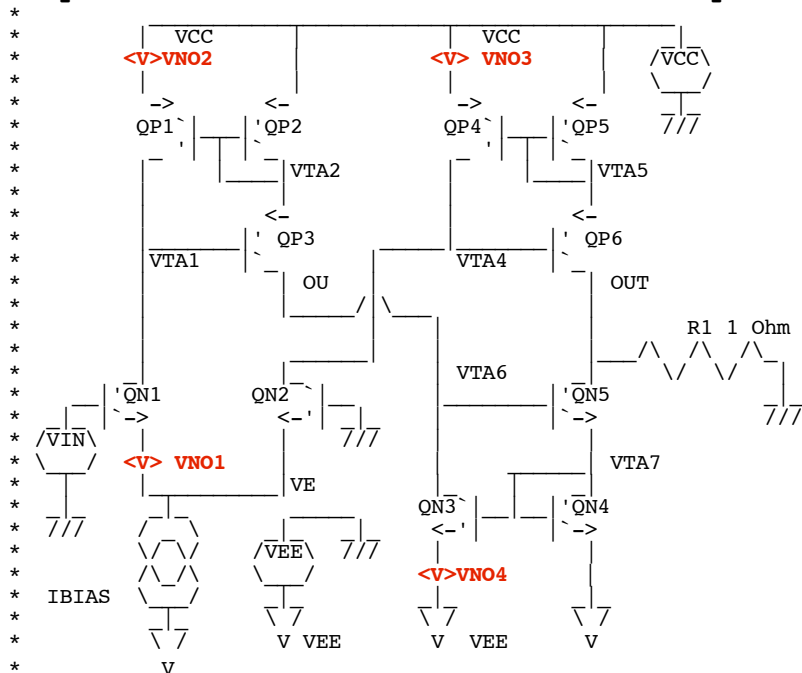
Rgm (vin/iout) 52360.3 Ohms
 Total referred to input noise_rthz **5.64871E-08 V**
 Vin maximum **0.0512053 V**
 Total output current noise_rthz 1.07881E-12 A
 Clipping +/- output current **9.7794E-07 A**
 Clipping 20-20KHz output S/N **73.1629 dB**

The LM3080 architecture provides a good standard way to measure differential input stages for Operational Transconductor Amplifiers (OTA). Signal to noise in particular tracks the square root of the supply current. Any comparisons made should respect that fact.

This simulation allows the introduction of expected process parameters like beta, offset, rbase, etc. Spice's "alter" state is used find the error contributions of each matched pair. In the LM3080 architecture, there are four matched pairs. If all the pair pairs had the same expected offset, then the total OTA noise and offset would be expected to be a factor of two worst than what comes from just the differential input stage.

The fact that differential input stage is distorted further reduces the offset and noise performance. If a the input signal is kept below the 1% level (+/-18mVpk), a 1.6mV worth of offset is only a factor of ten lower than this practical input level.

The distortion is often lowered by using a pre-distortion stage, which introduces other issues. Assuming the pre-distortion can be done without adding any offset and noise, then the input signal could be raised to the point just below where the output current clips at the 100% maximum output current. Without the pre-distortion stage, the 1% distortion level produces output current at 35% of the maximum output current.




```

*=====
VCC      VCC      0      DC      5
VEE      0      VEE      DC      5
VIN      VIN      0      DC      0
VNO1     VNO1     VE      DC      0
VNO2     VNO2     VCC     DC      0
VNO3     VNO3     VCC     DC      0
VNO4     VNO4     VEE     DC      0
IBIAS    VE      0      DC      1u
QN1      VTA1     VIN      VNO1     NPN1     1
QN2      VTA4     0      VE      NPN1     1
QP1      VTA1     VTA2     VNO2     PNP1     1
QP2      VTA2     VTA2     VCC     PNP1     1
QP3      VTA6     VTA1     VTA2     PNP1     1
QP4      VTA4     VTA5     VNO3     PNP1     1
QP5      VTA5     VTA5     VCC     PNP1     1
QP6      OUT     VTA4     VTA5     PNP1     1
QN3      VTA6     VTA7     VNO4     NPN1     1
QN4      VTA7     VTA7     VEE     NPN1     1
QN5      OUT     VTA6     VTA7     NPN1     1
R1       OUT     0      1

```

```

.control
setplot new
set thisData =      data
let $thisData =      0 * vector(18)
set thisData2 =      data2
let $thisData2 =      0 * vector(18)

```

```

let n=0
repeat 17
let      unknown.{$thisData2}[n] = n
let n=n+1
endrepeat

```

```

**0=====Define 1X_NPN_and_PNPL_and_RB_Offset=====

```

```

let      npnoff = .6m
let      pnpoff = 1.1m
let      rboff  = .005
let      unknown.{$thisData}[0] = npnoff
let      unknown.{$thisData}[1] = pnpoff
let      unknown.{$thisData}[2] = rboff
let      npnbf  = @NPN1[bf]
let      pnpbf  = @PNP1[bf]
let      npnrb  = @NPN1[rb]
let      pnprb  = @PNP1[rb]

```

```

echo      "NPN_offset  = $&nnpnoff  "
echo      "NPN_beta    = $&nnpnbf   "
echo      "NPN_rbase   = $&nnpnrb   "
echo      "PNPL_offset = $&pnpoff   "
echo      "PNPL_beta   = $&pnpbf    "
echo      "PNPL_rbase  = $&pnprb   "
echo      "Rbase_offset = $&rboff   "
echo      ""

```

```

**0=====Measure_With_No_Errors=====

```

```

op
let      unknown.{$thisData}[3] = out
let      ibb    = @ibias[dc]
let      icc    = -1*vcc#branch
echo      "Ibias      = $&ibb  "
echo      "Isupply    = $&icc  "
echo      ""

```

```

**1=====Measure_gm_At_1mV=====

```

```

alter      vin      dc =      1m
op
let      gm =      abs(out- unknown.{$thisData}[3]) /1m
let      unknown.{$thisData}[4] = gm
alter      vin      dc =      0

```

```

**2=====Measure_VNO1_effect=====

```

```

alter      vno1     dc =      1m
op
let      procOff = unknown.{$thisData}[0]
let      Tarea   = @qn1[area]
let      Ioutref = unknown.{$thisData}[3]
let      gm      = unknown.{$thisData}[4]
let      vingain = 1000*(out - Ioutref)/gm
let      pairOff = (1/Tarea)^.5*procOff
let      vinoff  = pairOff*vingain
let      unknown.{$thisData}[5] = vingain
let      unknown.{$thisData}[6] = vinoff

```

```

echo      "Pair      ProcOff  Area  Pair_Off  Gain2Vin  Vin_Offset "
echo      "QN1/QN2  $&procOff  $&Tarea  $&pairOff  $&vingain  $&vinoff"

**3=====Measure_VNO2_effect=====
alter     vno1    dc = 0
alter     vno2    dc = 1m
op
let       procOff = unknown.{ $thisData}[1]
let       Tarea   = @qp1[area]
let       Ioutref = unknown.{ $thisData}[3]
let       gm      = unknown.{ $thisData}[4]
let       vingain = 1000*(out - Ioutref)/gm
let       pairOff = (1/Tarea)^.5*procOff
let       vinoff  = pairOff*vingain
let       unknown.{ $thisData}[7] = vingain
let       unknown.{ $thisData}[8] = vinoff
echo      "QP1/QP2  $&procOff  $&Tarea  $&pairOff  $&vingain  $&vinoff"

**4=====Measure_VNO3_effect=====
alter     vno2    dc = 0
alter     vno3    dc = 1m
op
let       procOff = unknown.{ $thisData}[1]
let       Tarea   = @qp4[area]
let       Ioutref = unknown.{ $thisData}[3]
let       gm      = unknown.{ $thisData}[4]
let       vingain = 1000*(out - Ioutref)/gm
let       pairOff = (1/Tarea)^.5*procOff
let       vinoff  = pairOff*vingain
let       unknown.{ $thisData}[9] = vingain
let       unknown.{ $thisData}[10] = vinoff
echo      "QP4/QP5  $&procOff  $&Tarea  $&pairOff  $&vingain  $&vinoff"

**5=====Measure_VNO4_effect=====
alter     vno3    dc = 0
alter     vno4    dc = 1m
op
let       procOff = unknown.{ $thisData}[0]
let       Tarea   = @qn3[area]
let       Ioutref = unknown.{ $thisData}[3]
let       gm      = unknown.{ $thisData}[4]
let       vingain = 1000*(out - Ioutref)/gm
let       pairOff = (1/Tarea)^.5*procOff
let       vinoff  = pairOff*vingain
let       unknown.{ $thisData}[11] = vingain
let       unknown.{ $thisData}[12] = vinoff
echo      "QN3/QN4  $&procOff  $&Tarea  $&pairOff  $&vingain  $&vinoff"

**6=====Measure_total_offset=====
let       off1    = unknown.{ $thisData}[6]
let       off2    = unknown.{ $thisData}[8]
let       off3    = unknown.{ $thisData}[10]
let       off4    = unknown.{ $thisData}[12]
let       Vin_total = ( off1^2 + off2^2 + off3^2 + off4^2 )^.5

echo      " "
echo      "Total referred to input offset  $&Vin_total"
echo      " "

**7=====Measure_Maximum_Iout=====
alter     vno4    dc = 0
alter     vin     dc = -130m
op
let       unknown.{ $thisData}[12] = out
alter     vin     dc = 0
op

**9=====Calc_VNO1_effect_On_Noise=====
let       icc     = @qn1[ic]
let       Tarea   = @qn1[area]
let       rbb     = @NPN1[rb]/Tarea
let       NoiseR  = rbb + 13m/icc
let       NoiseP  = 4.07n*(2*NoiseR/1000)^.5
let       Gain2in = unknown.{ $thisData}[5]
let       VinNois = abs(NoiseP*Gain2in)
let       gm      = unknown.{ $thisData}[4]
let       IoutNo  = VinNois*gm
let       unknown.{ $thisData}[13] = VinNois
echo      "Pair , Ic , NoiseRes, Pair_Noise, Gain2Vin, VnoiseIn_rthz,OutNoise_rthz"
echo      "QN1/QN2, $&icc, $&NoiseR, $&NoiseP, $&Gain2in, $&VinNois, $&IoutNo"

**9=====Calc_VNO2_effect_On_Noise=====
let       icc     = @qp1[ic]
let       Tarea   = @qp1[area]

```



```

let      rbb      = @PNP1[rb]/Tarea
let      NoiseR   = rbb + 13m/icc
let      NoiseP   = 4.07e-9*(2*NoiseR/1000)^.5
let      Gain2in  = unknown.{ $thisData}[7]
let      VinNois  = abs(NoiseP*Gain2in)
let      gm       = unknown.{ $thisData}[4]
let      IoutNo   = VinNois*gm
let      unknown.{ $thisData}[14] = VinNois
echo     "QP1/QP2, $&icc, $&NoiseR, $&NoiseP, $&Gain2in, $&VinNois, $&IoutNo"

```

```

**!)=====Calc_VNO3_effect_On_Noise=====
let      icc      = @qp4[ic]
let      Tarea    = @qp4[area]
let      rbb      = @PNP1[rb]/Tarea
let      NoiseR   = rbb + 13m/icc
let      NoiseP   = 4.07e-9*(2*NoiseR/1000)^.5
let      Gain2in  = unknown.{ $thisData}[9]
let      VinNois  = abs(NoiseP*Gain2in)
let      gm       = unknown.{ $thisData}[4]
let      IoutNo   = VinNois*gm
let      unknown.{ $thisData}[15] = VinNois
echo     "QP4/QP5, $&icc, $&NoiseR, $&NoiseP, $&Gain2in, $&VinNois, $&IoutNo"

```

```

**!l=====Calc_VNO4_effect_On_Noise=====
let      icc      = @qn3[ic]
let      Tarea    = @qn3[area]
let      rbb      = @PNP1[rb]/Tarea
let      NoiseR   = rbb + 13m/icc
let      NoiseP   = 4.07e-9*(2*NoiseR/1000)^.5
let      Gain2in  = unknown.{ $thisData}[9]
let      VinNois  = abs(NoiseP*Gain2in)
let      gm       = unknown.{ $thisData}[4]
let      IoutNo   = VinNois*gm
let      unknown.{ $thisData}[16] = VinNois
echo     "QN3/QN4, $&icc, $&NoiseR, $&NoiseP, $&Gain2in, $&VinNois, $&IoutNo"

```

```

let      off1     = unknown.{ $thisData}[13]
let      off2     = unknown.{ $thisData}[14]
let      off3     = unknown.{ $thisData}[15]
let      off4     = unknown.{ $thisData}[16]
let      Vin_T    = ( off1^2 + off2^2 + off3^2 + off4^2 )^.5
let      Out_T    = Vin_T*gm
let      ibb      = @ibias[dc]
let      ibb      = unknown.{ $thisData}[12]
let      SigNo    = 20*log( ibb/(Out_T*141) ) -3

```

```

echo     " "
echo     "Total referred to input noise_rthz  $&Vin_T V"
echo     "Total output current noise_rthz    $&Out_T A"
echo     "Clipping +/- output current         $&ibb A"
echo     "Clipping 20-20KHz output S/N       $&SigNo dB"

```

```

echo     " "

```

```

.endc

```

```

*===== models Using Near Perfect Transistors =====
.model  NPN1  NPN(  BF = 200  VAF = 100  rb=400 )
.model  PNP1  PNP(  BF = 30   VAF = 150  rb=900 )
*=====

```

```

.end

```