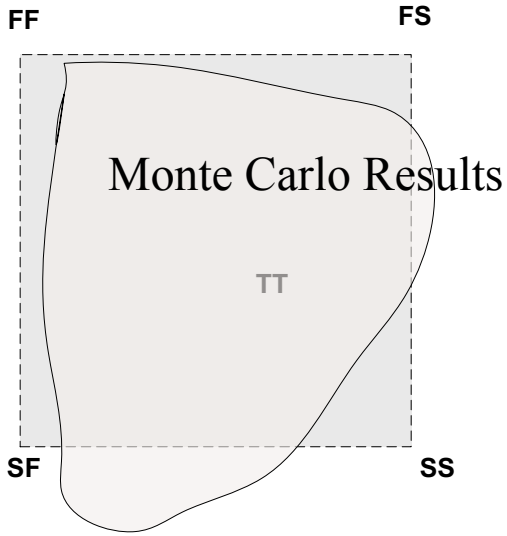
Monte Carlo Analysis

吕志军

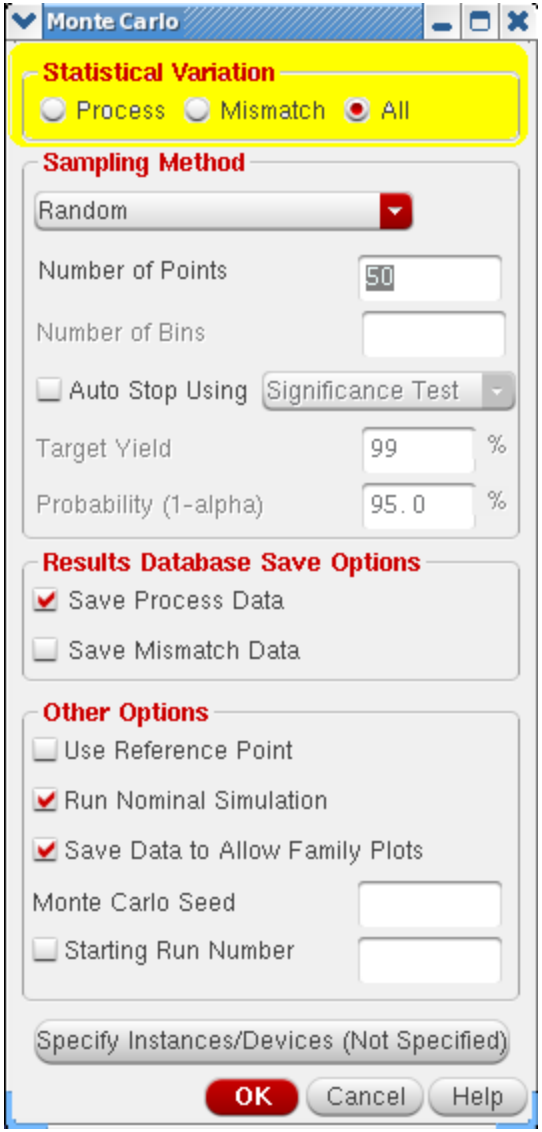
leolv@cadence.com

Monte Carlo VS Corner



1. MC gives an idea of yield (or distribution). Corner is much more extreme and will give a too pessimistic result.
2. MC is more accurate and can get mismatch information.

Process & Mismatch: Definition




```
parameters rr=1000
statistics {
process {
vary rr dist=gauss std=20
}
}
```

```
parameters rr=1000
statistics {
mismatch {
vary rr dist=gauss std=20
}
}
```

The distributions specified in the **process** block are sampled once per Monte Carlo iteration and are typically used to represent batch-to-batch or process variations.

The distributions specified in the **mismatch** block are sampled on a per subcircuit instance basis and are typically used to represent device-to-device mismatch for devices on the same chip.

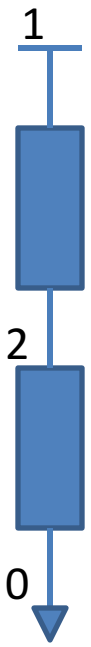
In the case where the same parameter is subject to **both process** and **mismatch** variations, the sampled **process** value becomes the mean for the mismatch random number generator for that particular parameter.

Process & Mismatch: Example

1. 如果一个参数，例如电阻值rr,只有process的变化，如下：

```
parameters rr=1000

statistics {
process {
vary rr dist=gauss std=20 percent=yes
}
}

r1 1 2 resistor r=rr
r2 2 0 resistor r=rr
v0 1 0 vsource dc=2 type=dc
```

MC分析结果，r1和r2均会变化，但变化同步，即V(2)始终为1V；



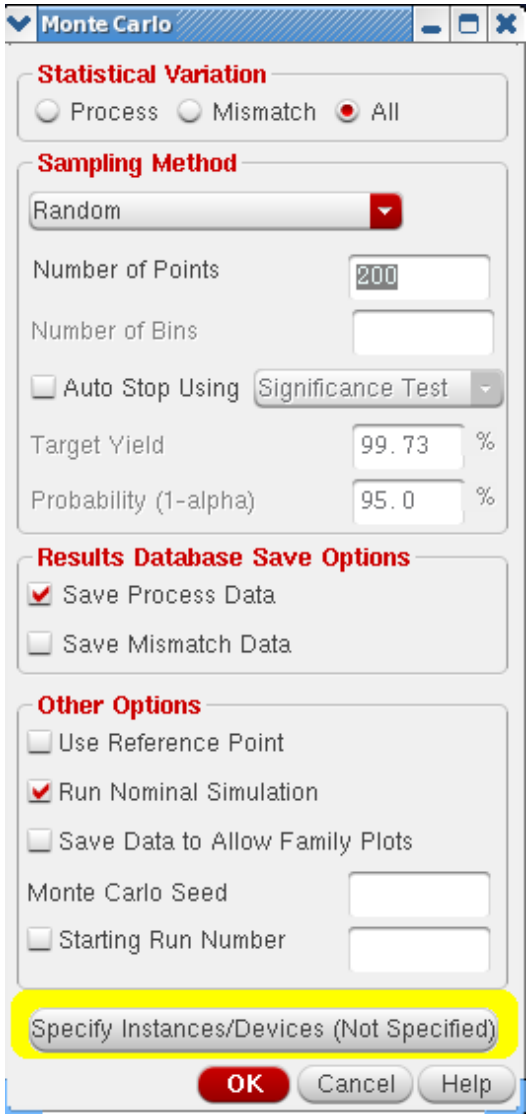
2. 如果rr只有mismatch的变化，如：

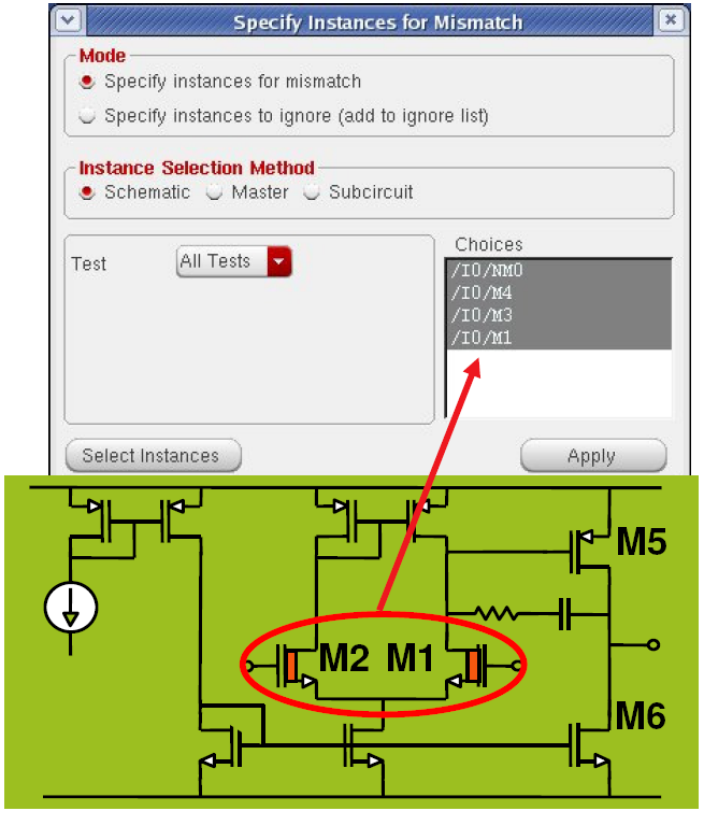
```
parameters rr=1000

statistics {
mismatch {
vary rr dist=gauss std=20 percent=yes
}
}

subckt res in out
r1 in out resistor r=rr
ends

x1 1 2 res
x2 2 0 res
```

MC结果，x1(r1)和x2(r2)变化不同步，即V(2)在1V上下分布取值；

mismatch: 参数rr处于subckt或inline subckt中，才会取值有变化；如果rr处于类似于1中本征器件中，则不会有mismatch的变化。而process的参数处于subckt(inline subckt)还是本征器件中，process的变化都会发生，且变化同步。

Mismatch: All or Selected?





Default, **Mismatch** will be run on **All Instances/Devices**.

Furthermore, **Mismatch** of only **Selected Device/Instance** could be considered or ignored in MC.

Spice Global&Local

1. Global corner/local corner

Spice model 里有以下几类corner

```
.LIB GlobalMC_LocalMC_MOS_MOSCAP
.LIB LocalMCOnly_MOS_MOSCAP
.LIB TTGlobalCorner_LocalMC_MOS_MOSCAP
```

一般我们跑process&mismatch的时候会选GLobalmc_localmc的model。

a)如果只跑process:要选globalmc_localmc还是localmc_only? 这两个在spice model里的参数只有个别项不一样，大部分是一样的。

b) globalmc_localmc和localmc_only 对于仿真来说有啥不一样?

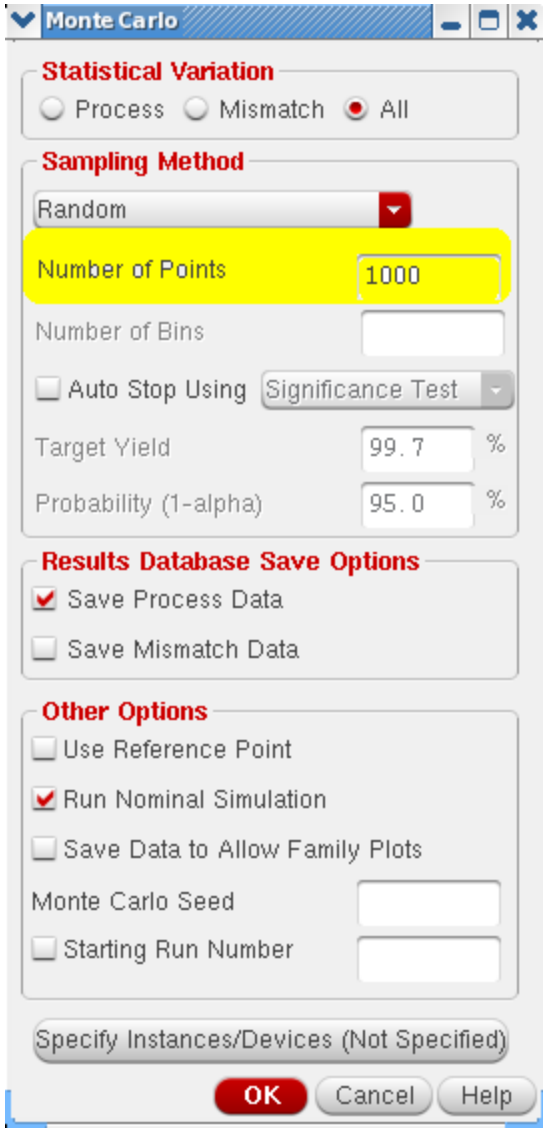
c) 如果只跑mismatch 是否需要定义globalcorner? 也就是用TTGloabalconer_localmc这类的spice model吗?

1. a) if run process only, need select GlobalMC_LocalMC.

b) Global_Local includes both process variation and local mismatch variation. LocalMC_only only has local device mismatch variation, it will ignore wafer-to-wafer and die-to-die's variation.
For simulation, the P = P(nominal) + deltaP(global) + deltaP(local), you will only see the deltaP(local)'s influence in simulation results when use LocalMC_only.

c) For mismatch no need to define global corner.

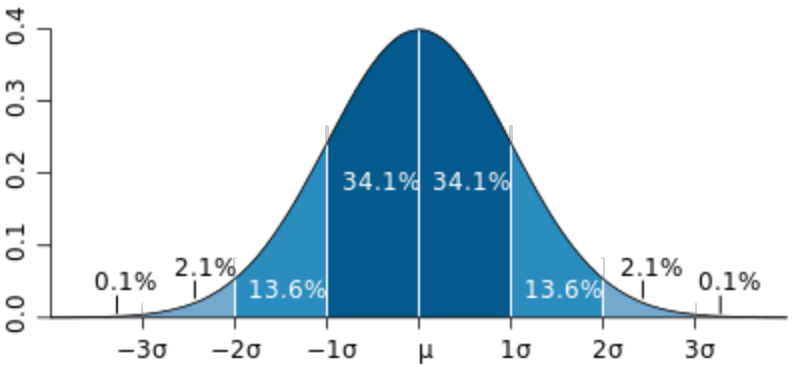
How Many Runs Needed: Basically…



Basically, The number of runs depends on the accuracy you want to achieve on your yield estimates.
If you want to know whether the yield is over 99.7% (3 sigma), 1000 simulations needed…

How Many Runs Needed: However…

However, in many case, **mean** and **sigma** are of interest.



若随机变量服从一个平均值为u、标准偏差为σ的正态分布，记为：

$$X \sim N(\mu, \sigma^2),$$

则其概率密度函数为

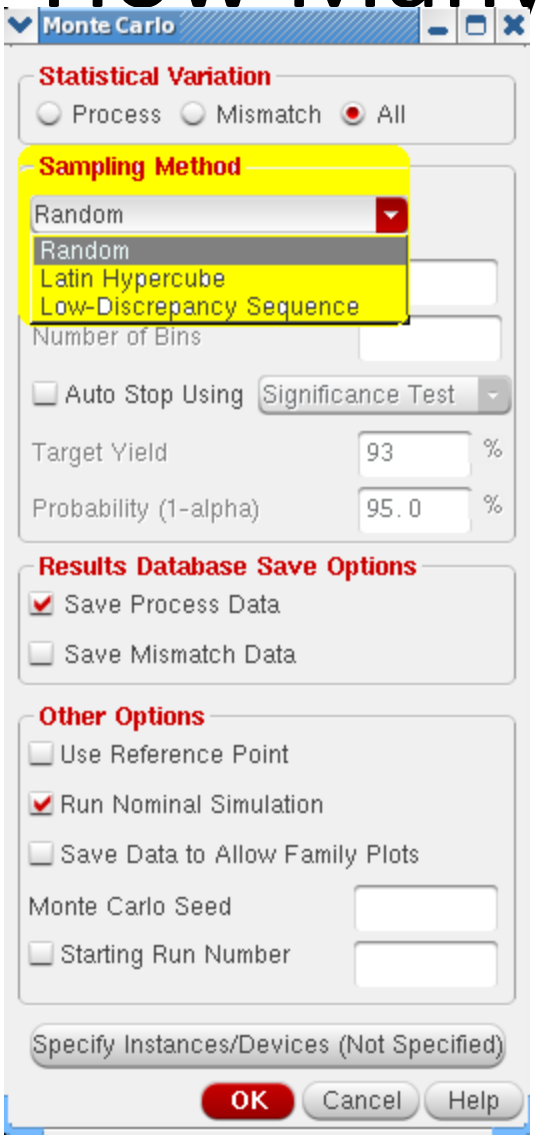
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

For normal distribution:

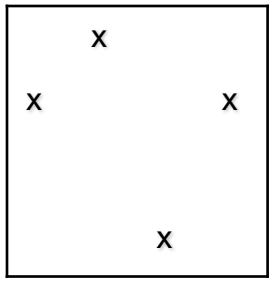
Distribution of X	Probability
[Mean-1*Sigma, Mean+1*Sigma]	68%
[Mean-2*Sigma, Mean+2*Sigma]	95%
[Mean-3*Sigma, Mean+3*Sigma]	**99.7%**

If 3-sigma range is within specification, **99.7% yield** is guaranteed!

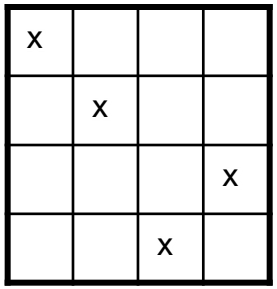
How Many Runs Needed: So…



So, other sampling method could be used to the same distribution with less sampling point.



Standard Random Sampling



LHS sampling (need to store samples)

Sampling Method	Number of Runs	Sigma
Random	100	16.5687
Random	1000	20.1577
LHS	100	19.8644
LHS	1000	19.7874

LDS:
Mean: slower than LHS; Sigma: faster than LHS.
Could use Target Yield Stopping Criteria.

MC model:Spectre VS Hspice

剩下一个问题就是, 我们用spectre格式的model跑variation=all, sigma比spice格式的小, 请帮忙再看下:

LHS, 100 samples:

T28 model (variation=all)	GMC+LMC	LMC
hspice/toplevel.l	Mean=698.4m 1sigma=5.29m	698.0m 5.026m
spectre/toplevel.scs	Mean=701.4m 1sigma=3.59m	701.0m 3.91m

Random, 1001 samples:

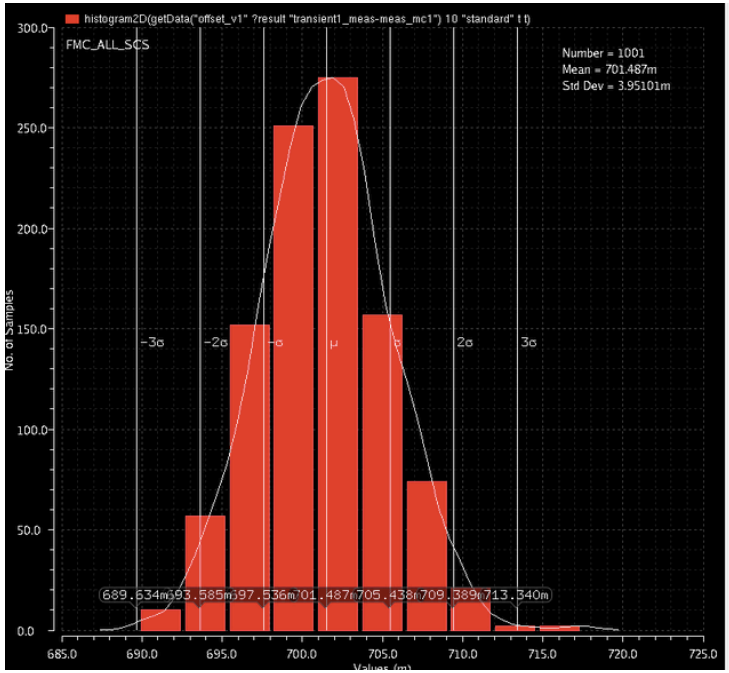
Hspice/toplelvel.l: 701.217m/3.99849m

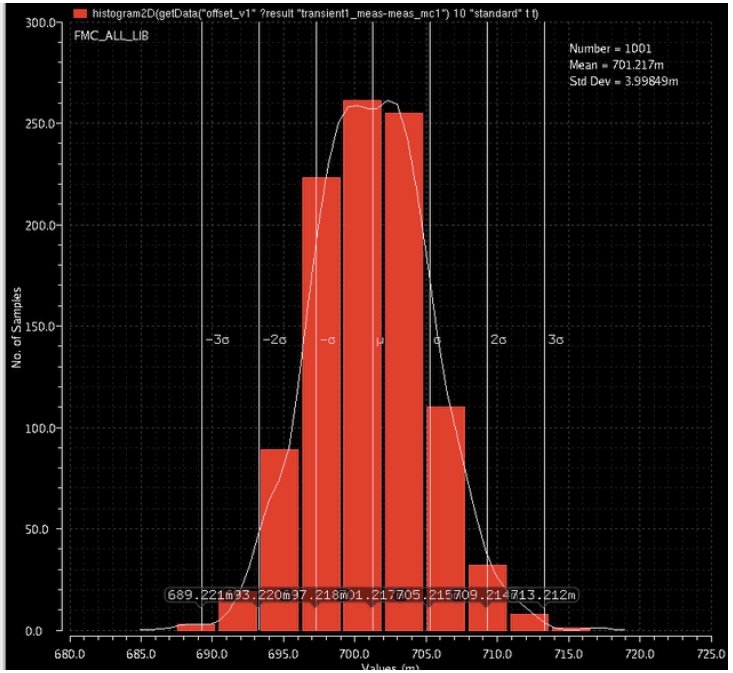
Spectre/toplevel.scs: 701.487m/3.95101m

从结果上看，.l 的model对lhs的算法支持不好。

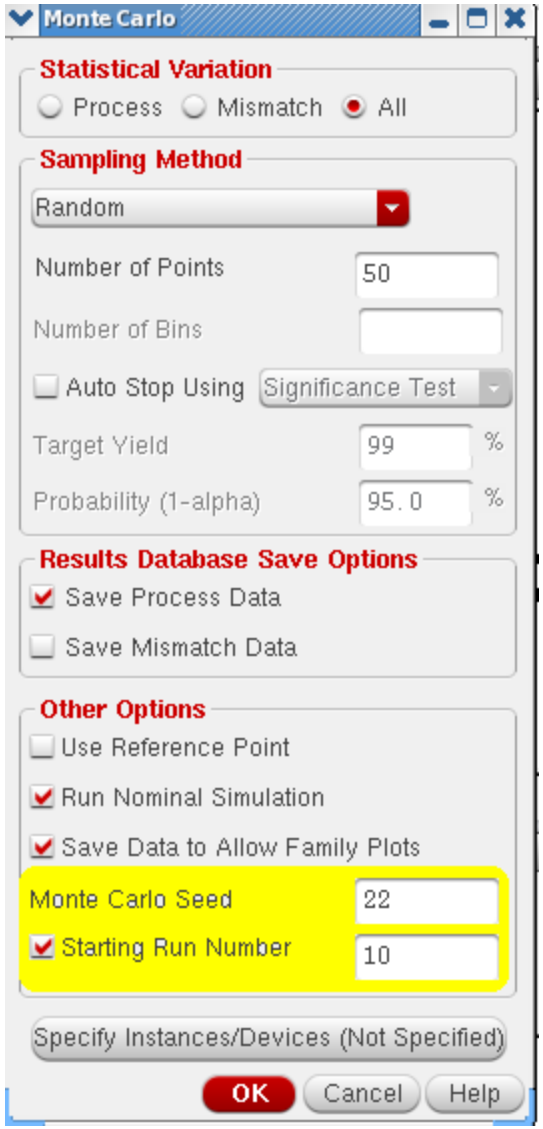
.scs的model使用lhs后的结果和跑1000个点的结果匹配较好。

所以还是建议使用.scs的model。





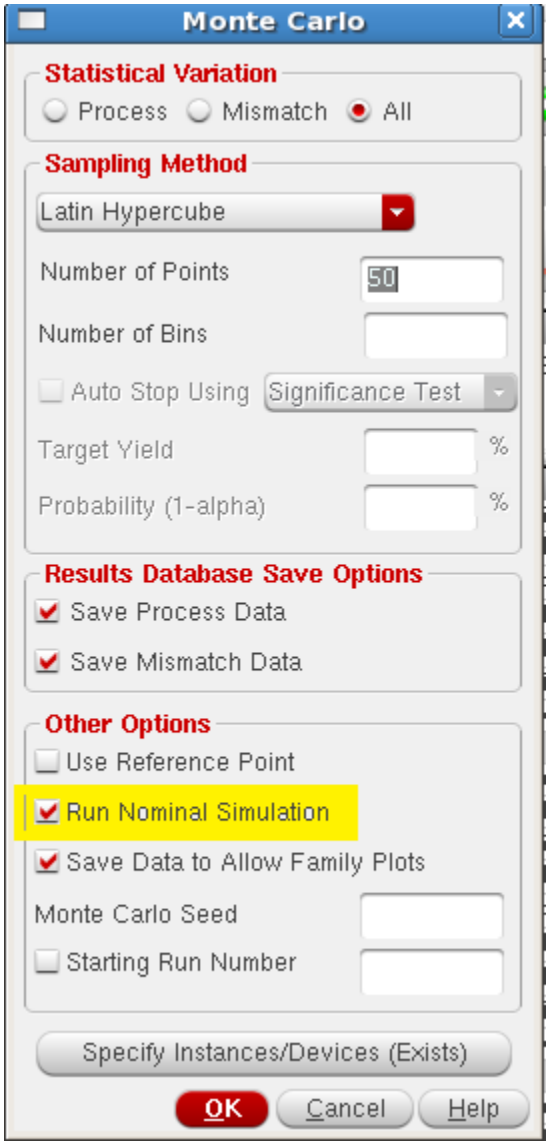
How to Reproduce: Seed & firstrun



By specifying the same seed, you can reproduce a previous experiment. If you do not specify a seed, the value 12345 is used.

By specifying the first iteration number and the same value for seed, you can reproduce a particular run or sequence of runs from a previous experiment (for example, to examine an outlier case in more detail.)

Run Nominal Simulation



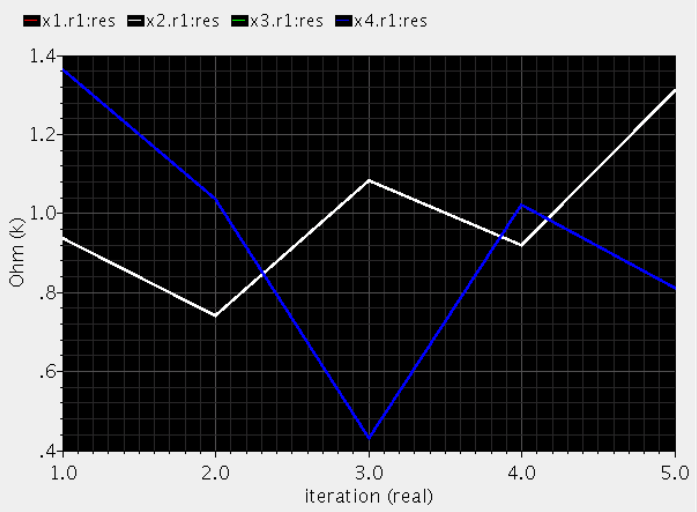
donominal={yes,no}(defaults to yes)

Controls whether Spectre should perform a nominal run before starting the main Monte Carlo loop of iterations. If any errors are encountered during the nominal run (for example, convergence problems, incorrect expressions, and so on) then Spectre issues an appropriate error message and **immediately abandons** the Monte Carlo analysis.

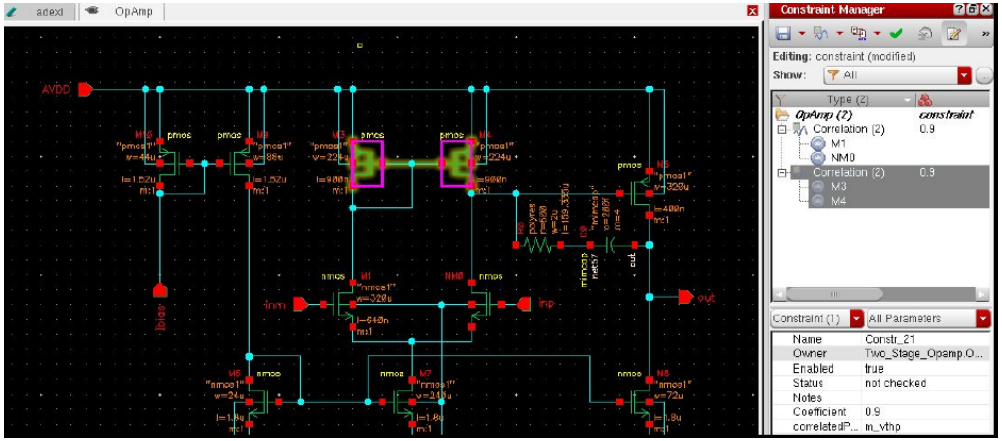
If set to no, Spectre runs only the Monte Carlo iteration, and does not perform nominal analysis. If any errors are encountered during the Monte Carlo iterations, Spectre issues a warning and continues with the next iteration of the Monte Carlo loop.

How to Correlate?: cc=[-1:1]

```
parameters rr=1000
statistics {
mismatch {
vary rr dist=gauss std=20 percent=yes
}
}

statistics {
correlate dev=[x1 x2] param=[rr] cc=1
correlate dev=[x3 x4] param=[rr] cc=1
}

subckt res in out
r1 in out resistor r=rr
ends

x1 1 0 res
x2 2 0 res
x3 3 0 res
x4 4 0 res
```



Res in X1 and X2 are exactly the same;
Res in X3 and X4 are exactly the same;



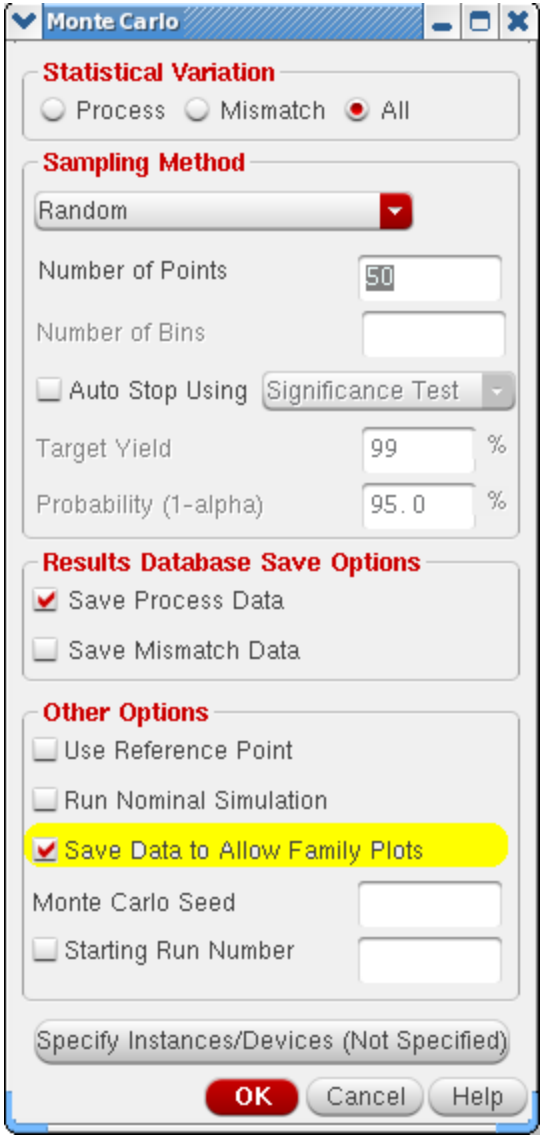
Correlation could also be set in Schematic:
(M3, M4), coefficient = 0.9;

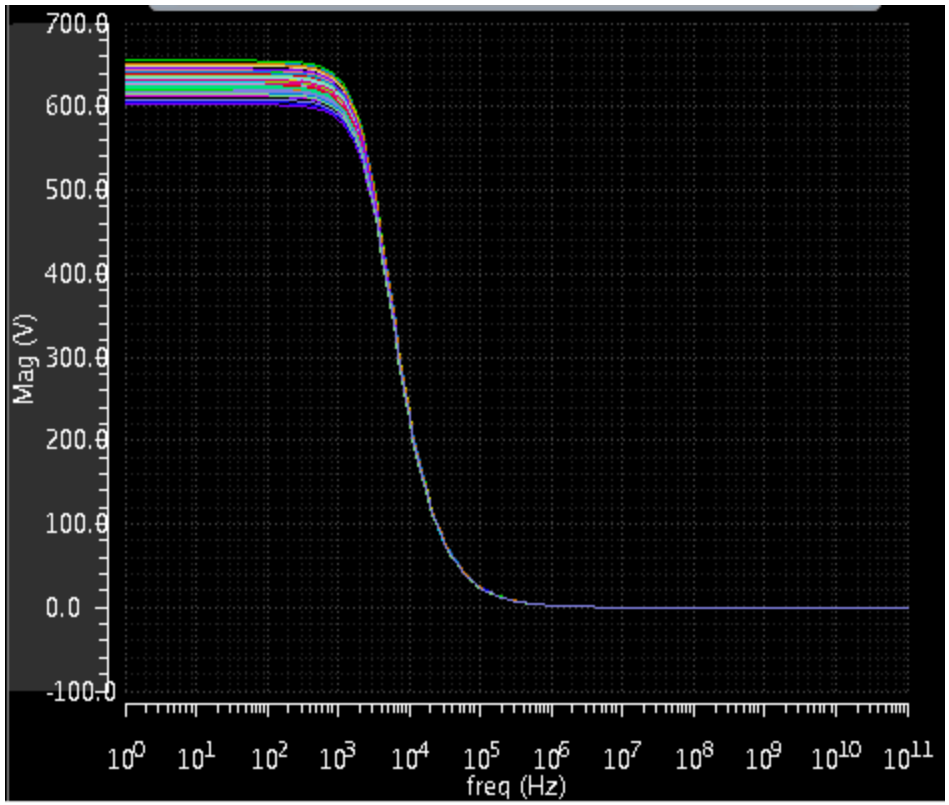
Performing Monte Carlo Analysis with LHS

```
mc1 montecarlo numruns=100 seed=1 variations=all sampling=lhs\
donominal=yes savefamilyplots=yes {
tran tran stop=500n
dcOp dc oppoint=logfile
export Vout=oceanEval("VDC(\"net32\")")
}
```

Parameters	Descriptions
numruns=100	Perform 100 monte carlo simulation runs
seed	Starting seed for the random number generator
variations=process	Specifies the typeof variations to consider
sampling=lhs	Method of statistical sampling to apply. Possible values are *standard*, l*hs* and *orthogonal*.
donominal=yes	Controls whether or not to perform the initial run that checks validity of the saved data.
savefamilyplots=no	This options allows you to save psf data for every iteration

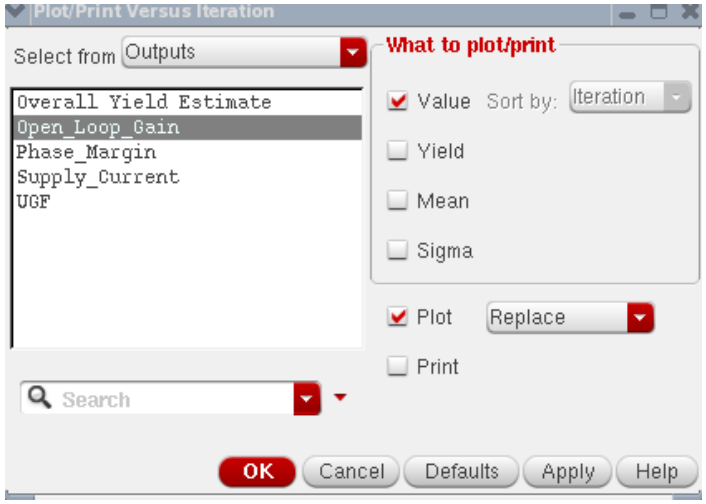
Results: Family Plot

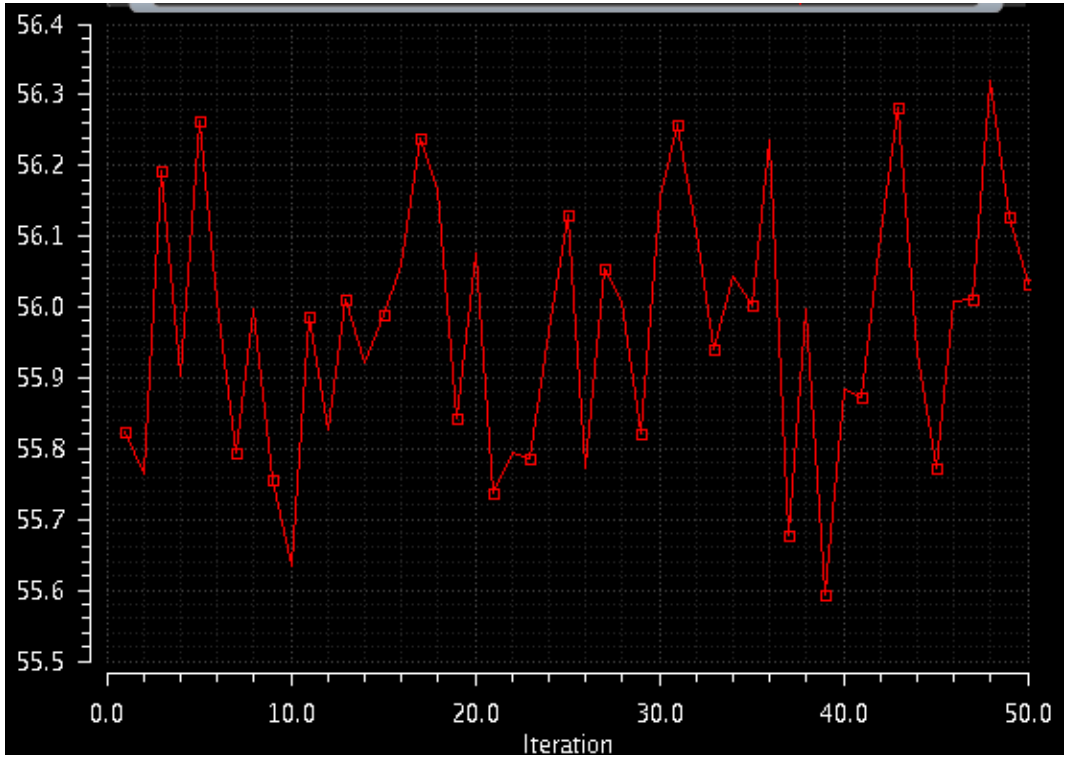




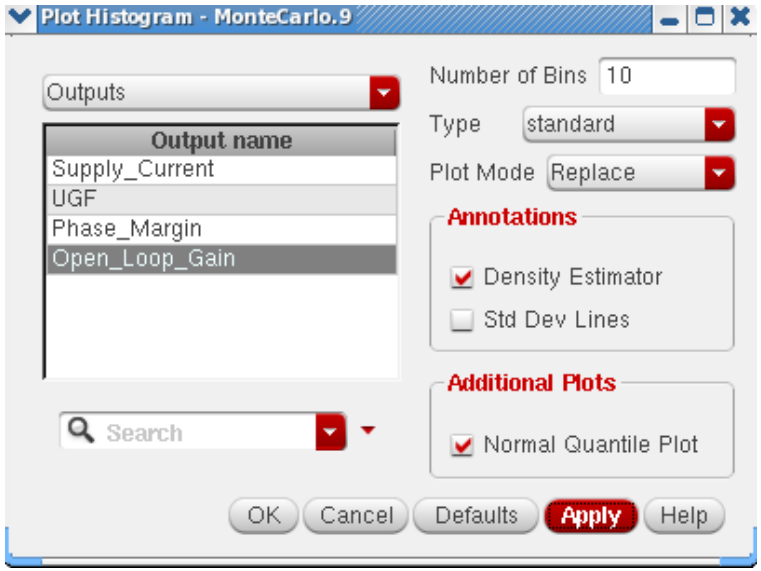
Results: Plot/Print Versus Iteration

You can print/plot outputs and parameters Vs iterations. In addition, you can also plot or print the mean and sigma values for outputs.

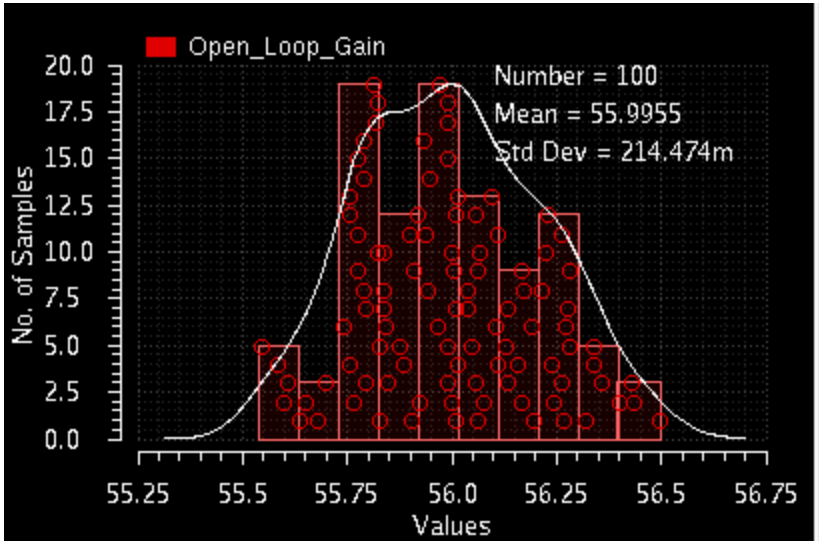


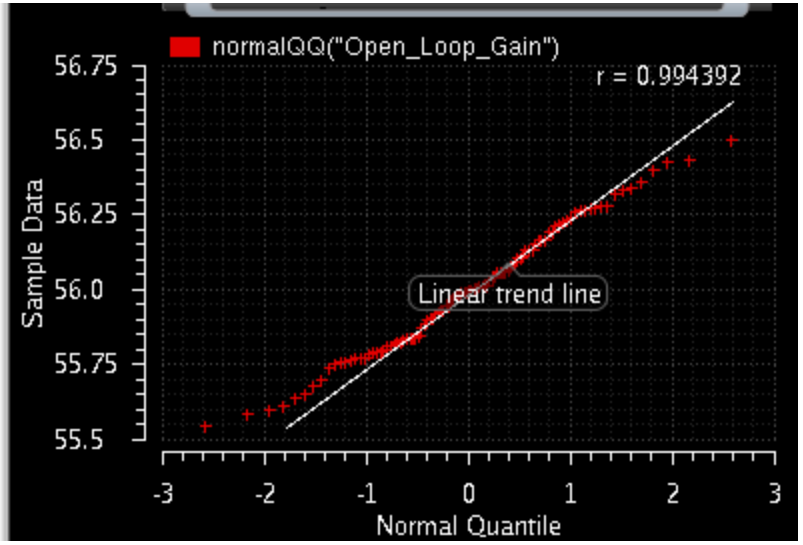


Results: Histograms



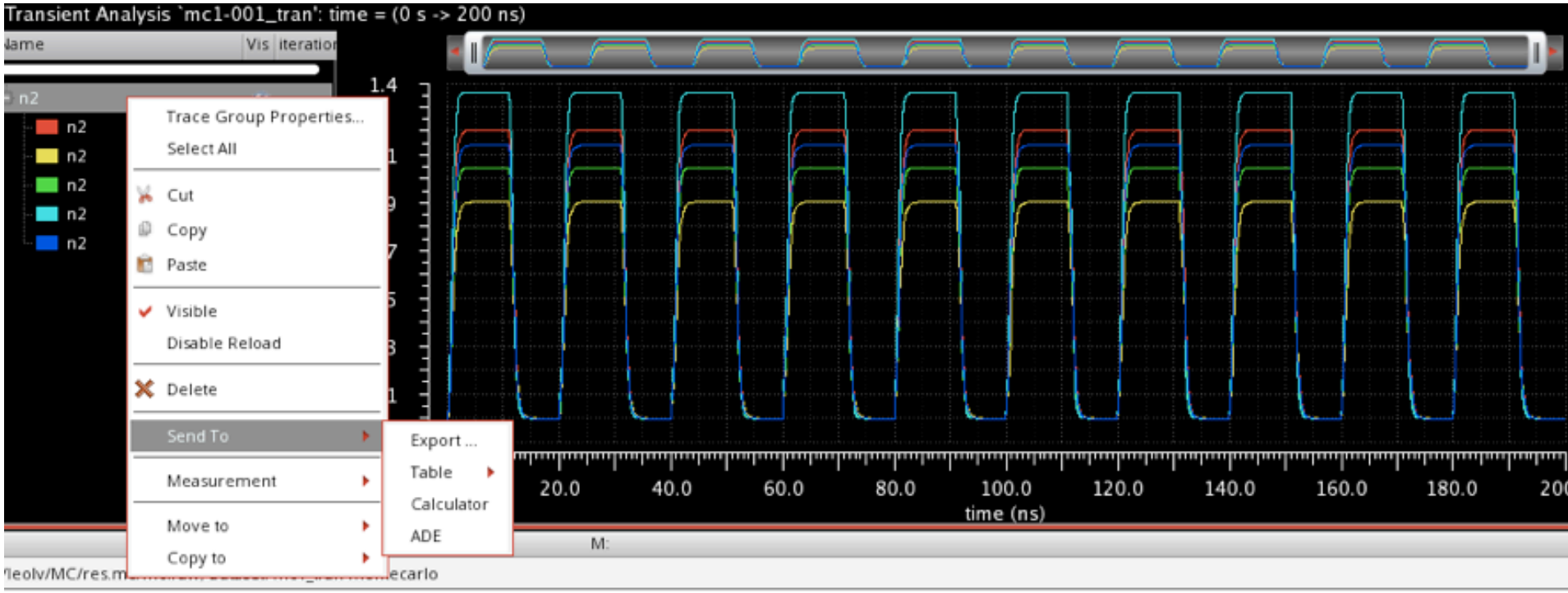
Histograms and Normal Quantile for the outputs and statistical parameters from Monte Carlo results can be ploted.





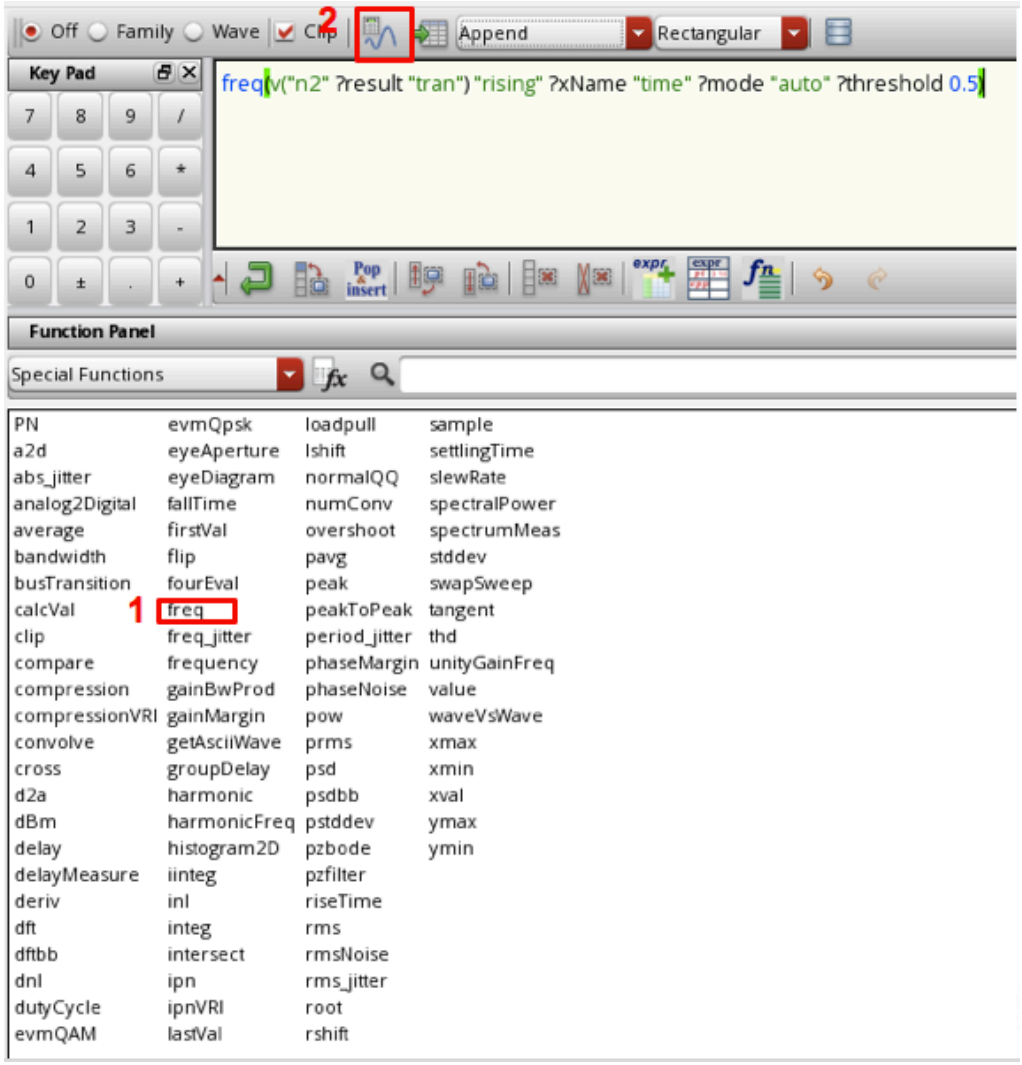
Example1: Frequency VS time

1. 右键点选MC跑出的同一节点的一组信号，选择send to: Calculator



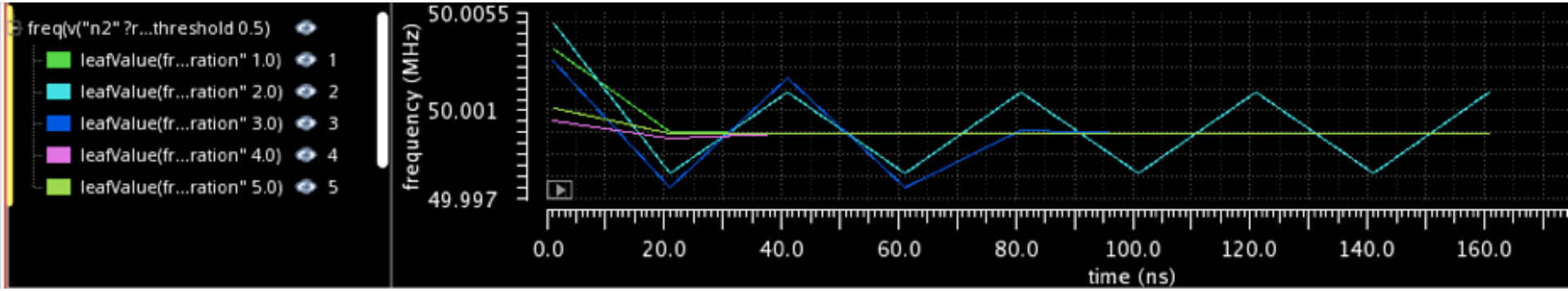
Example1: Frequency VS time

2. 在calculator中，给选中的信号添加函数”freq”,如下“红字1”所标函数：在点击图标”evaluate the buffer…”,如下图红字2所示图表。



Example1: Frequency VS time

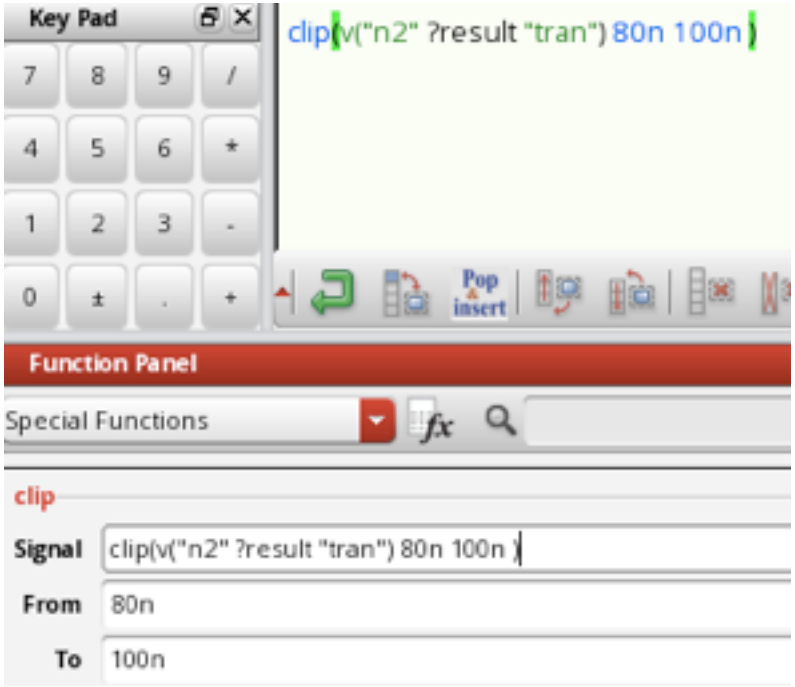
3. 得到5条频率随时间变化的波形：



Example2: Frequency VS index

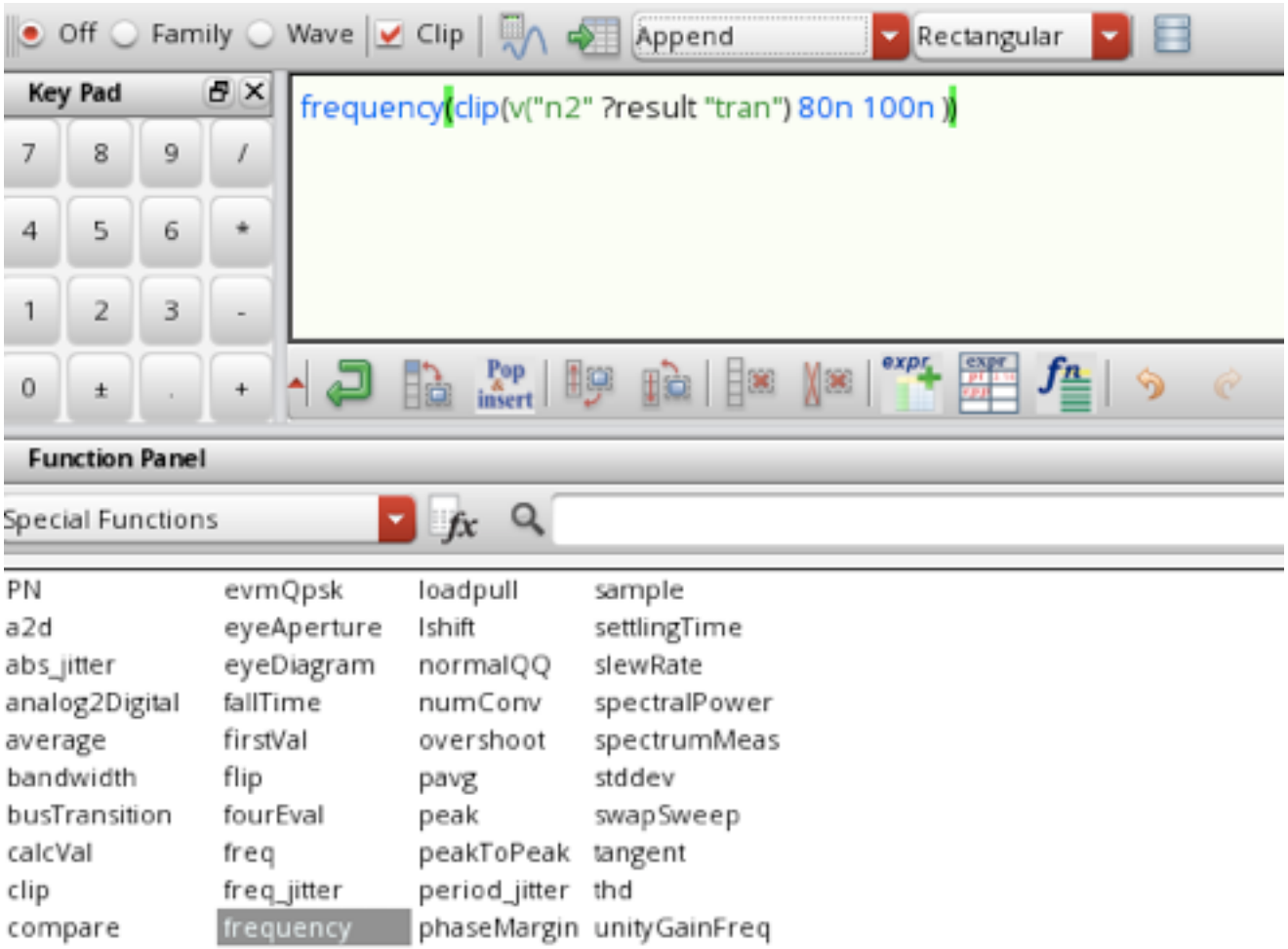
1. 右键点选MC跑出的同一节点的一组信号，选择send to: Calculator

2. 在calculator中，点选函数clip,并通过clip设置time区间



Example2: Frequency VS index

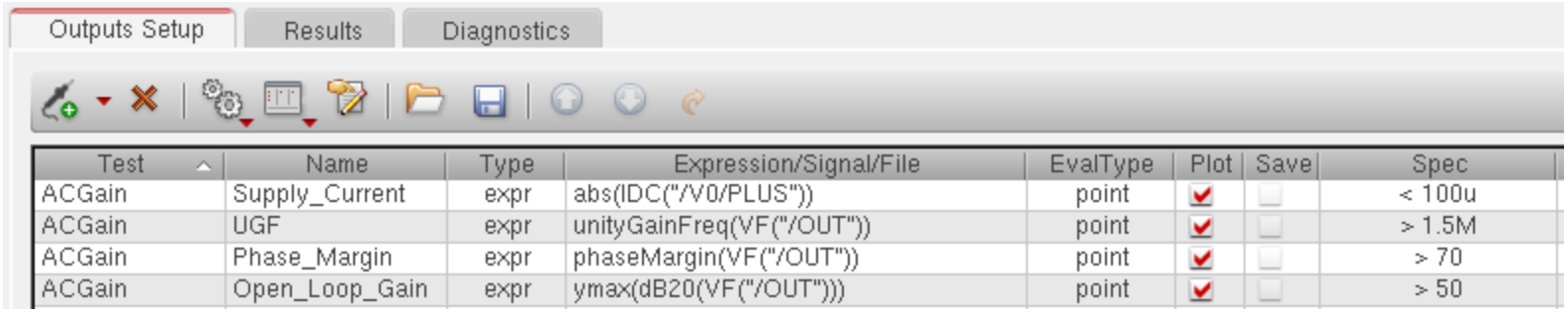
3. 依然在calculator中，点选”frequency”函数（不是freq）



ADE

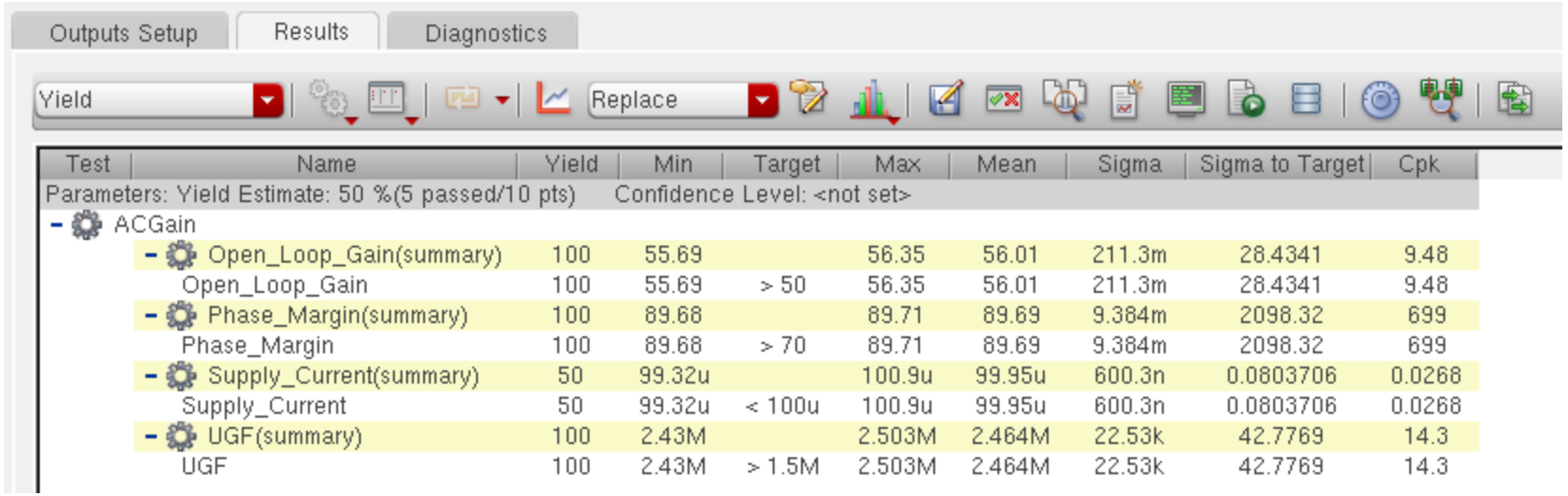
Specification & Yield

Set Specification:



Test	Name	Type	Expression/Signal/File	EvalType	Plot	Save	Spec
ACGain	Supply_Current	expr	abs(IDC("/V0/PLUS"))	point	☑	☐	< 100u
ACGain	UGF	expr	unityGainFreq(VF("/OUT"))	point	☑	☐	> 1.5M
ACGain	Phase_Margin	expr	phaseMargin(VF("/OUT"))	point	☑	☐	> 70
ACGain	Open_Loop_Gain	expr	ymax(dB20(VF("/OUT")))	point	☑	☐	> 50

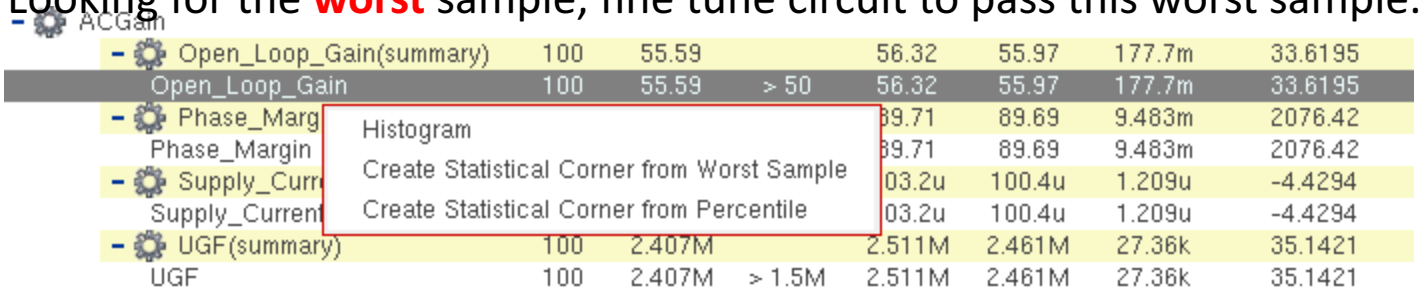
Get Yield based on Specification



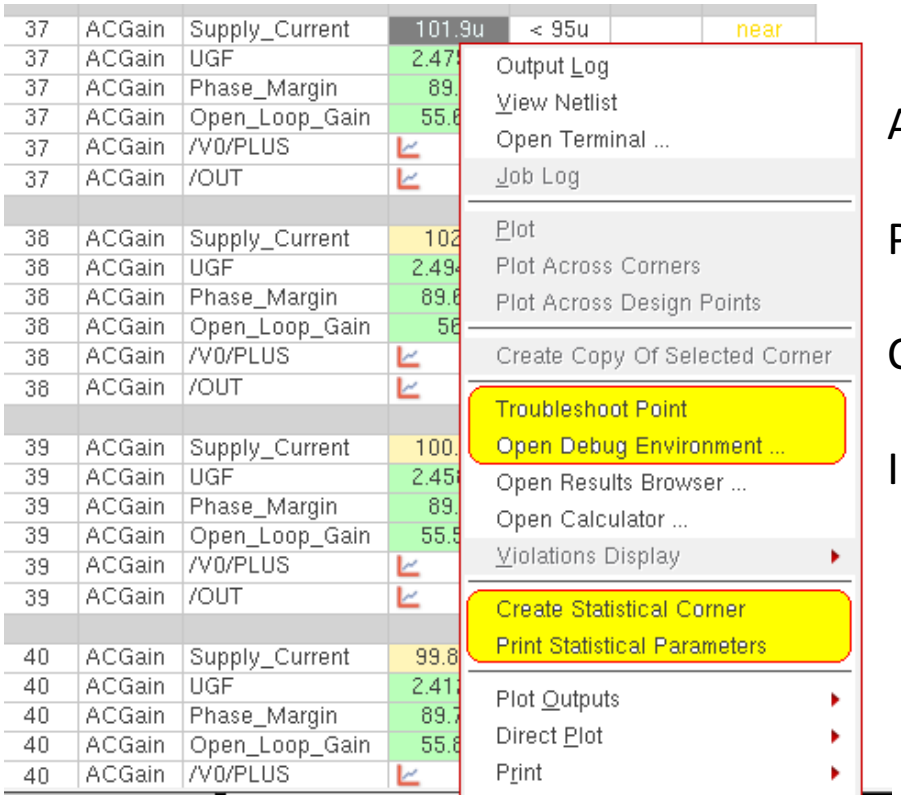
Test	Name	Yield	Min	Target	Max	Mean	Sigma	Sigma to Target	Cpk
Parameters: Yield Estimate: 50 %(5 passed/10 pts)		Confidence Level: <not set>							
ACGain									
Open_Loop_Gain(summary)		100	55.69		56.35	56.01	211.3m	28.4341	9.48
Open_Loop_Gain		100	55.69	> 50	56.35	56.01	211.3m	28.4341	9.48
Phase_Margin(summary)		100	89.68		89.71	89.69	9.384m	2098.32	699
Phase_Margin		100	89.68	> 70	89.71	89.69	9.384m	2098.32	699
Supply_Current(summary)		50	99.32u		100.9u	99.95u	600.3n	0.0803706	0.0268
Supply_Current		50	99.32u	< 100u	100.9u	99.95u	600.3n	0.0803706	0.0268
UGF(summary)		100	2.43M		2.503M	2.464M	22.53k	42.7769	14.3
UGF		100	2.43M	> 1.5M	2.503M	2.464M	22.53k	42.7769	14.3

How to Simplify: Worst or Specified

Looking for the **worst** sample, fine tune circuit to pass this worst sample.



Percentile is most least worst sample number.



Anyone of MC samples could be **Specified** to:

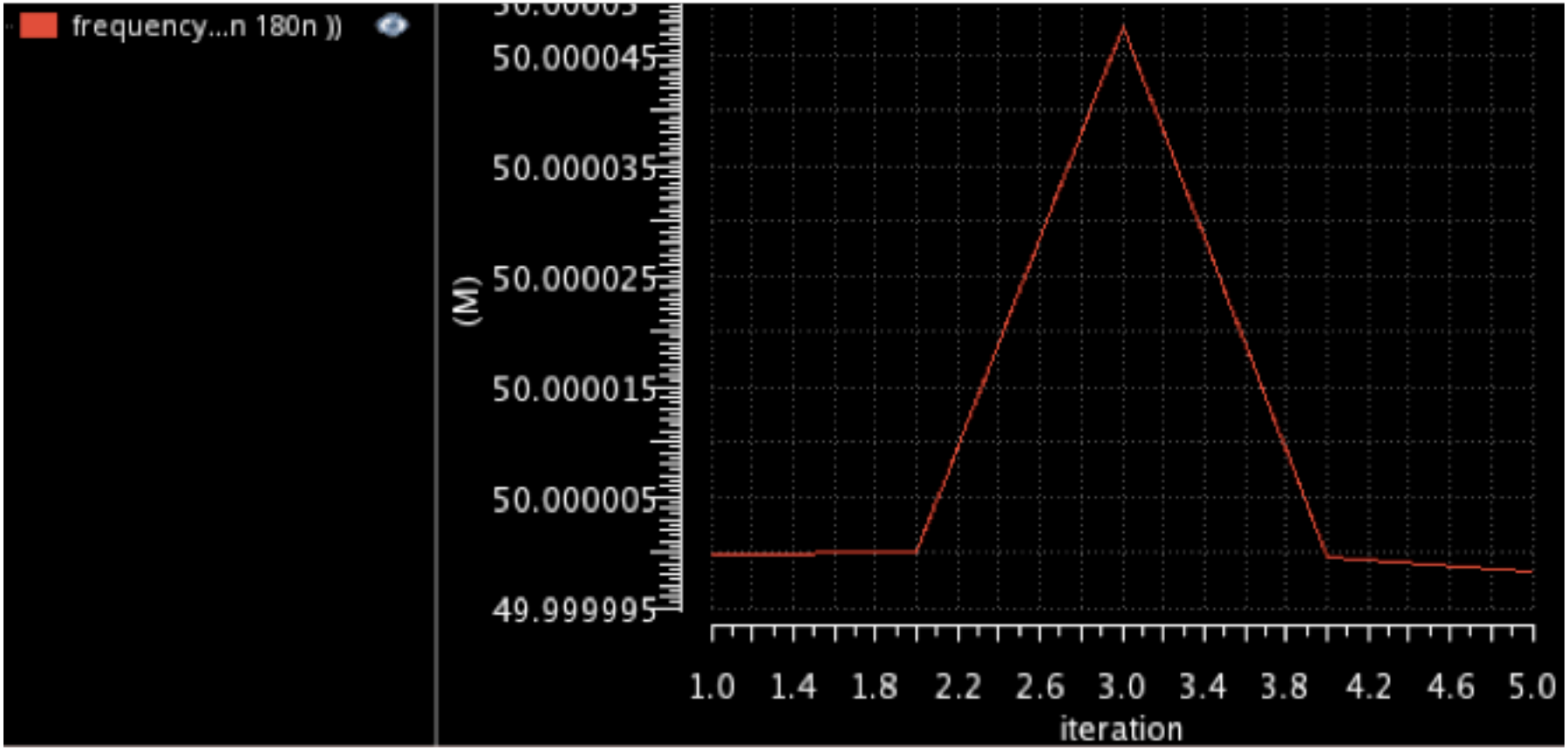
Print the statistical parameter of the corner;

Create this MC sample to a single coner;

In-line debug this corner.

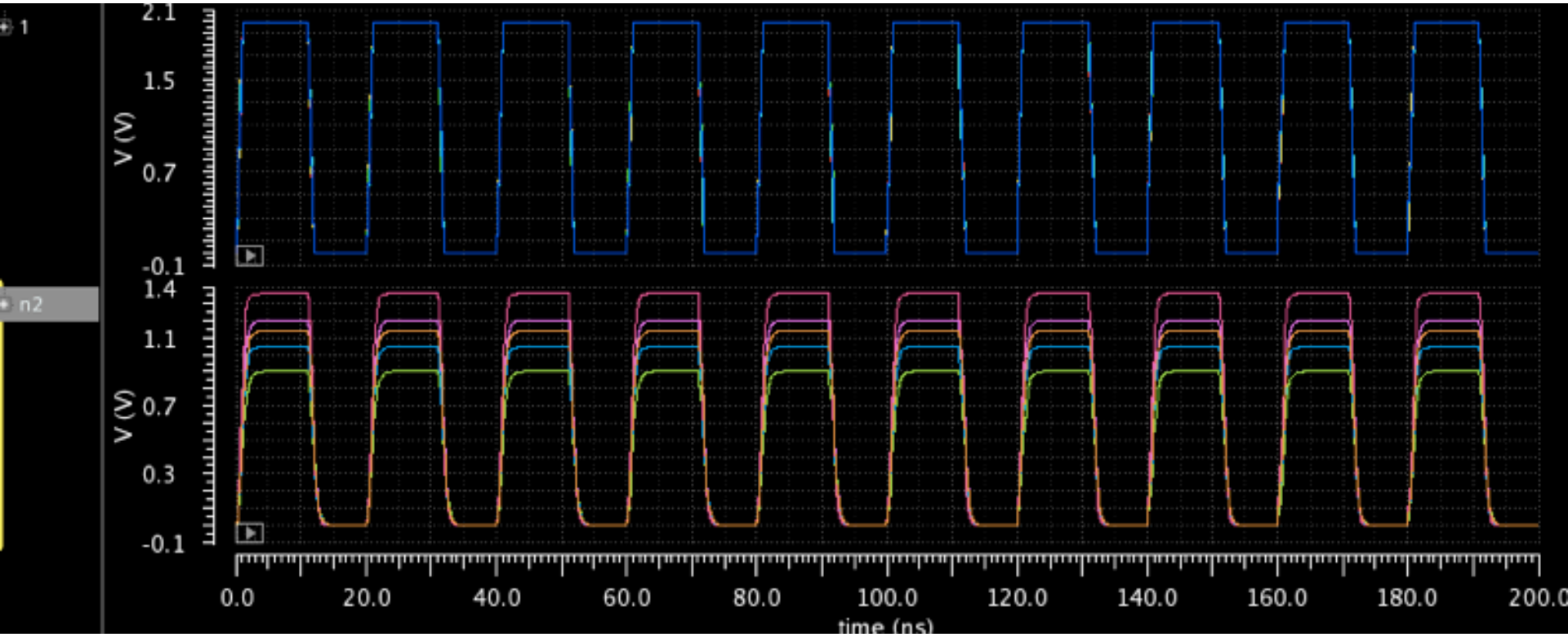
Example2: Frequency VS index

4. 在calculator中点击图标”evaluate the buffer…”,得到如下横坐标是index,纵坐标是所选区间平均频率的曲线：



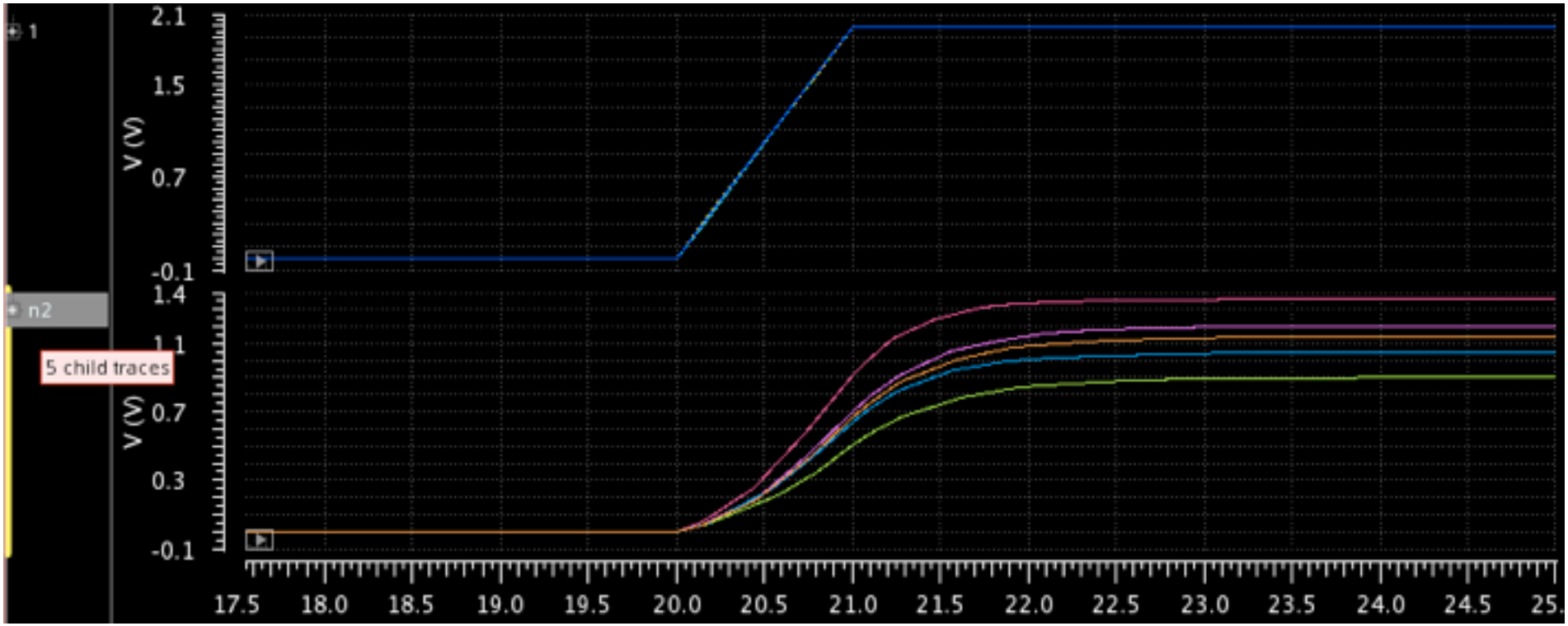
Example 3: Delay Histogram

1. MC仿真跑出2个信号：v(1), v(n2)，如下：



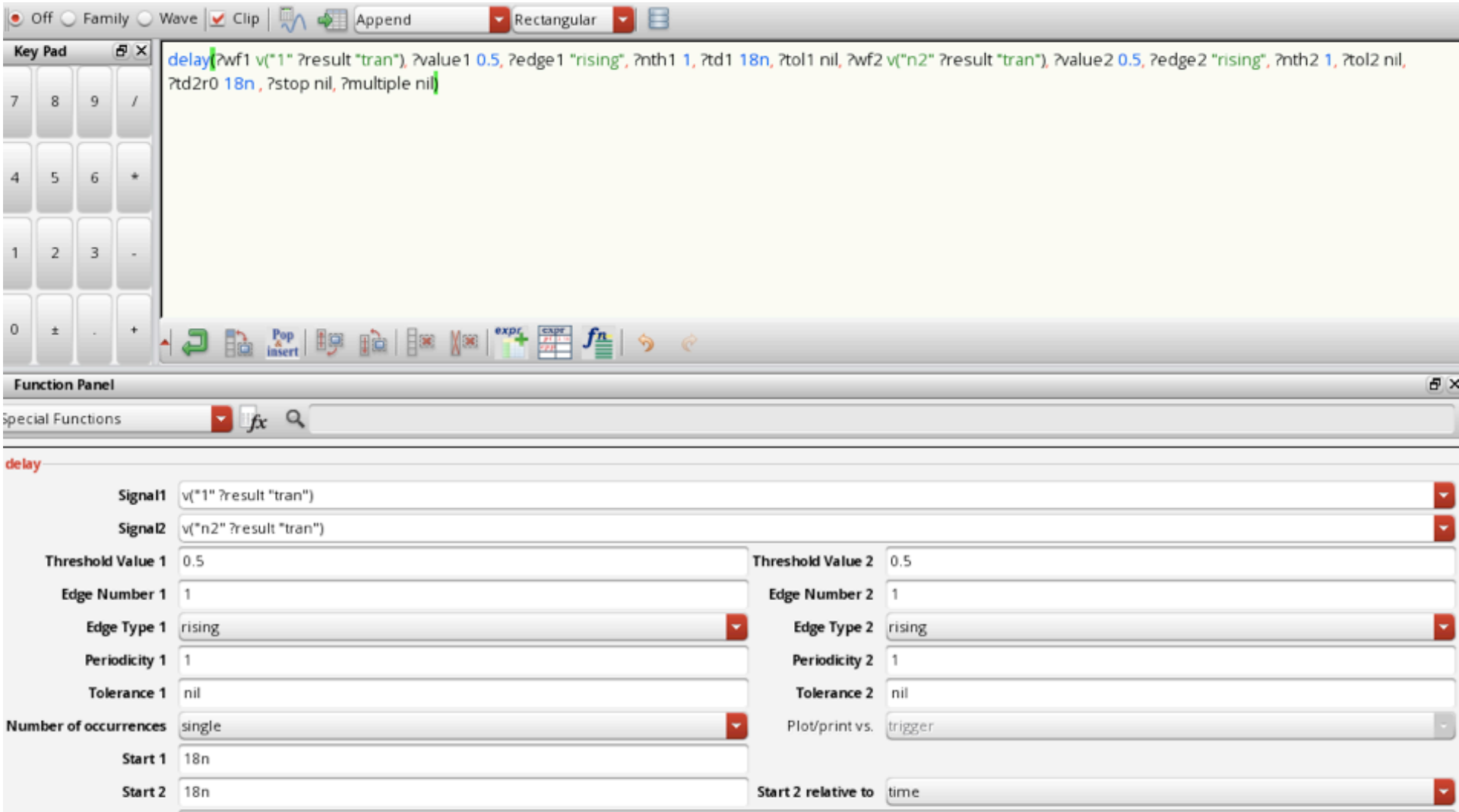
Example 3: Delay Histogram

2. 现在想看18ns后，v(1)和v(n2)的第一个上升沿之间的delay：



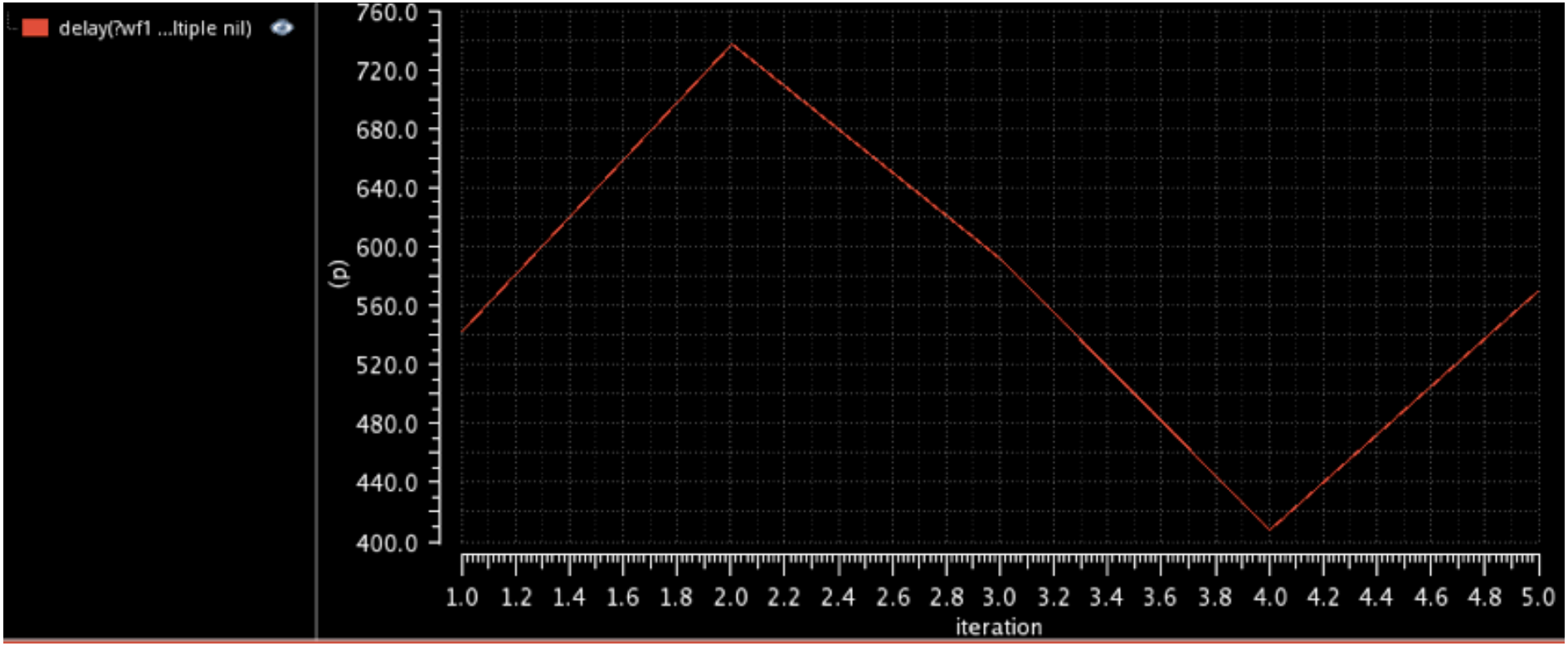
Example 3: Delay Histogram

3. 右键点选v(1)或v(n2)，选择send to: Calculator,然后调用函数delay如下：



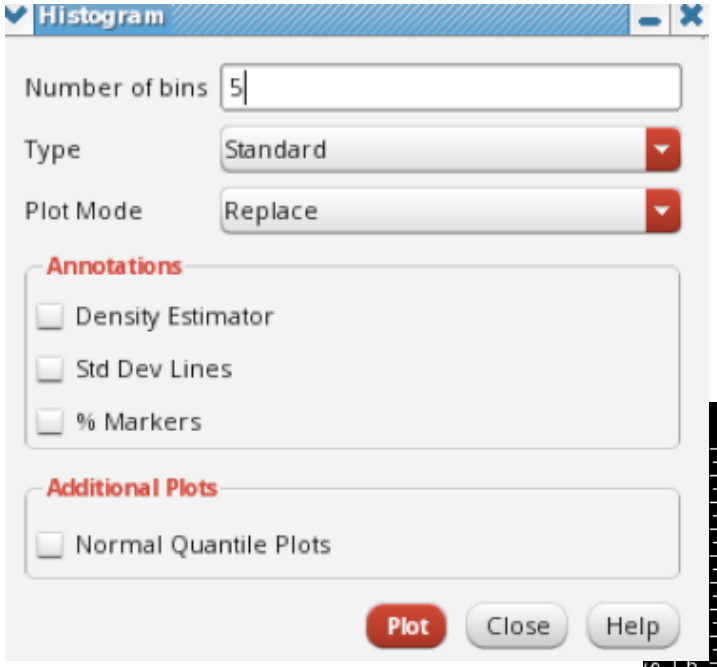
Example 3: Delay Histogram

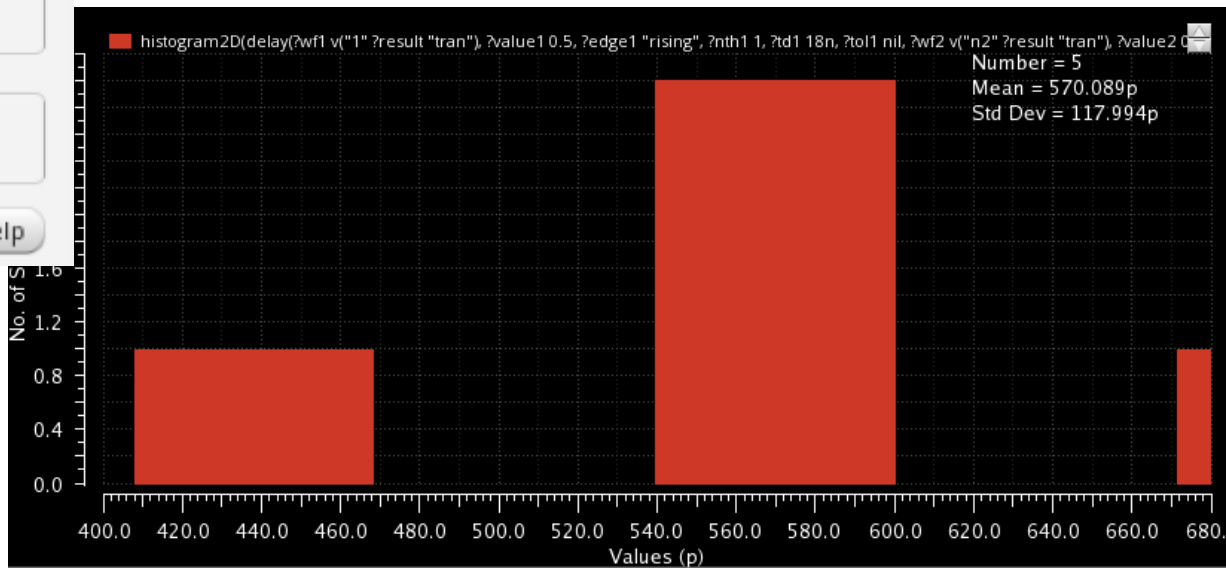
点击calculator上的图标”evaluate the buffer…  ，得到横坐标是index,纵坐标是delay的波形如下：



Example 3: Delay Histogram

5. 右键上图中的信号，选择measurment->Histogram 设置number of bins，如设为5





END