折叠共源共栅放⼤器设计实验报告

181180058 李禄轶

**摘要**：折叠共源共栅运算放⼤器（Folded Cascode Operational Amplifier），具有较⾼的输出阻抗，较

⼤的电压输出摆幅，应⽤⼴泛，在 CMOS 图像传感器，模数转换器ADC中是相当重要的模块电路。本次课程设计通过理论计算和仿真分析，完成了满⾜性能指标要求的折叠共源共栅放⼤器设计。

[**折叠共源共栅放⼤器设计实验报告**](#_bookmark0)

1. [实验环境](#_bookmark1)
2. [基本要求及参数](#_bookmark2)
   1. [性能指标要求](#_bookmark4)
   2. [MOS管参数测量](#_bookmark3)[3 放⼤器理论计算](#_bookmark5)
3. [电流漏的设计及仿真](#_bookmark6)
4. [整体电路仿真以及性能测试](#_bookmark7)
   1. [放⼤电路搭建](#_bookmark8)
   2. [开环电压增益Av测量](#_bookmark9)
   3. [带宽增益积GB测量](#_bookmark10)
   4. [相位裕量Phase margin测量](#_bookmark11)
   5. [共模抑制⽐CMRR测量](#_bookmark12)
   6. [共模输⼊范围ICMR测量](#_bookmark13)
   7. [压摆率SR测量](#_bookmark14)
   8. [电源抑制⽐PSRR测试](#_bookmark15)
   9. [Pdiss测量](#_bookmark16)[6 实验总结](#_bookmark17)

[7 附: parameter.c 源代码](#_bookmark18)

# 实验环境

硬件环境：VMware fusion 12虚拟机

操作系统：Red Had Enterprise Linux AS Release 4，内核版本2.6.9

软件版本：Cadence 5.1.0

# 基本要求及参数

## 性能指标要求

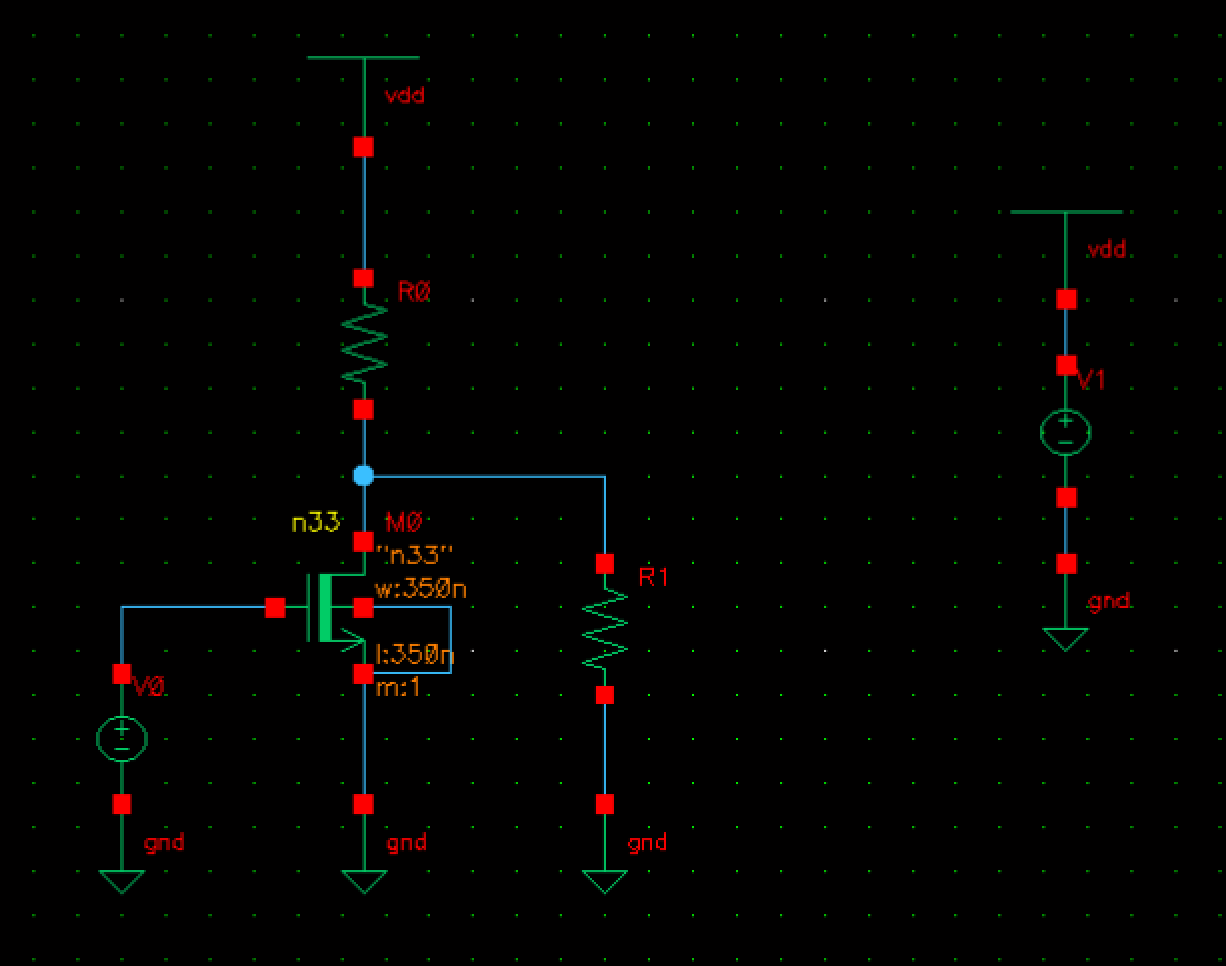
|  |  |
| --- | --- |
| **指标名称** | **要求数值** |
| Vdd | 3.3V |
| Av | >3000 |
| GB | 3MHz |
| Phase margin | >50 |
| SR | >5V/ s |
| ICMR | 1.1V to 2.5V |
| Vout range | 0.6V to 2.7V |
| Pdiss | 5mW |
| CL | 10pF |

* 1. **MOS管参数测量**

本次课程设计中，所有的MOS管均采⽤SMIC018库中的n33和p33。为了进⾏理论分析，我们需要 知道 这些⼯艺参数。

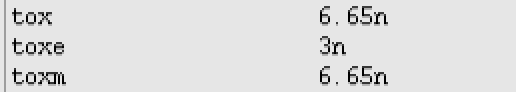
1. **NMOS管**

搭建⼀个简单的共源级放⼤器电路，输⼊电压的交流分量幅值500mV，直流分量设为变量Vin。在 仿真中，选择DC分析，变量Vin从0到3.3V扫描，保存静态⼯作点。成功Run之后，点击Results->Print-

>Model Parameters，在图上选择NMOS管，即可看到打印在窗⼝上的参数。

**图1 NMOS管参数测量电路**

打印结果如下：



**图2 NMOS管⼯艺参数**

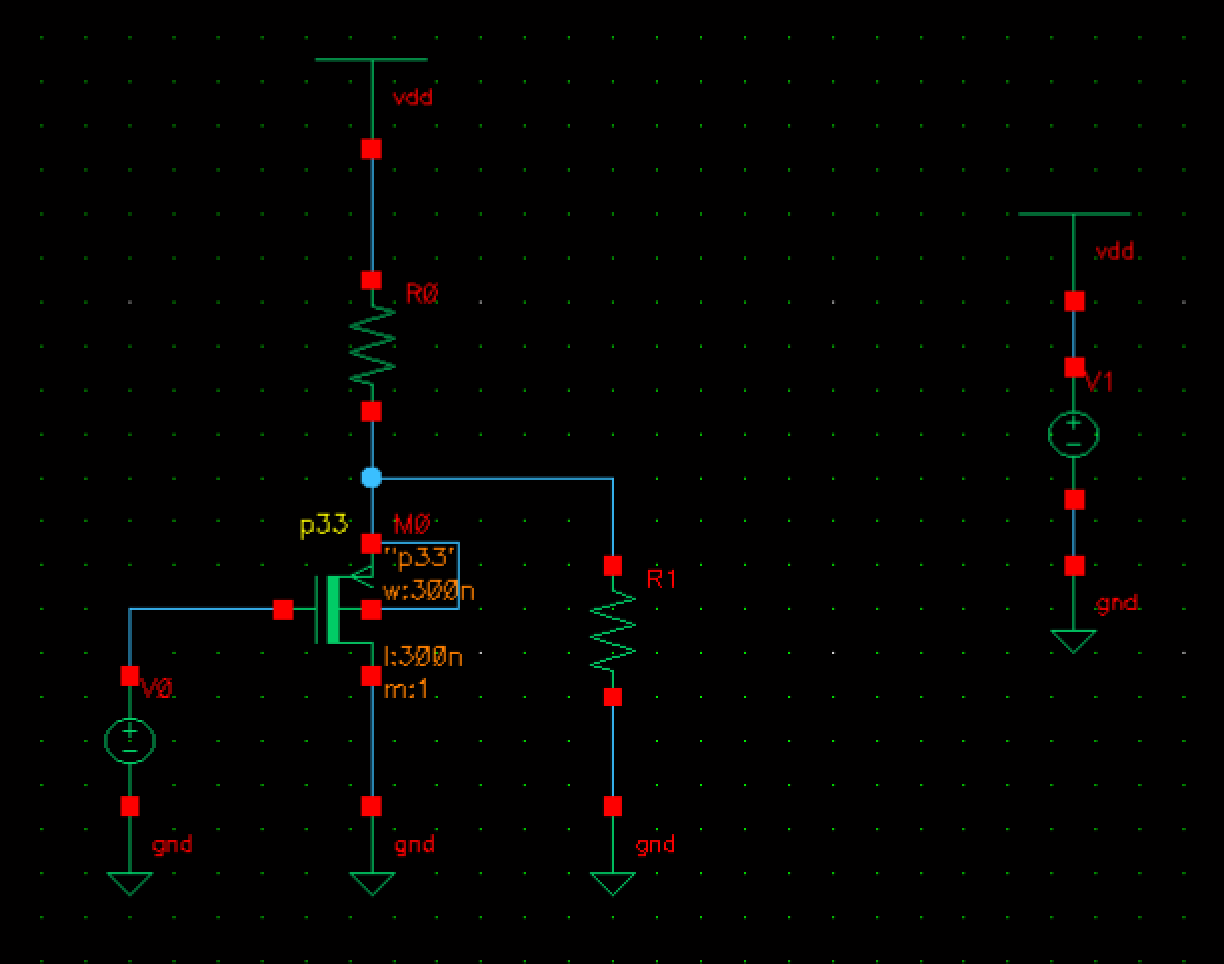
对于  和  之间选择，这⾥可能名称有误，但需要选择较⼤的那个（氧化层总的有效厚度），该 处为6.65nm。由此，我们可以计算出 ：





1. **PMOS管**

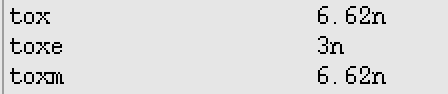
对于PMOS管，同样搭建测试电路。

打印结果如下：

计算 ：

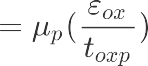
**图3 PMOS管参数测量电路**

**图4 PMOS管⼯艺参数**

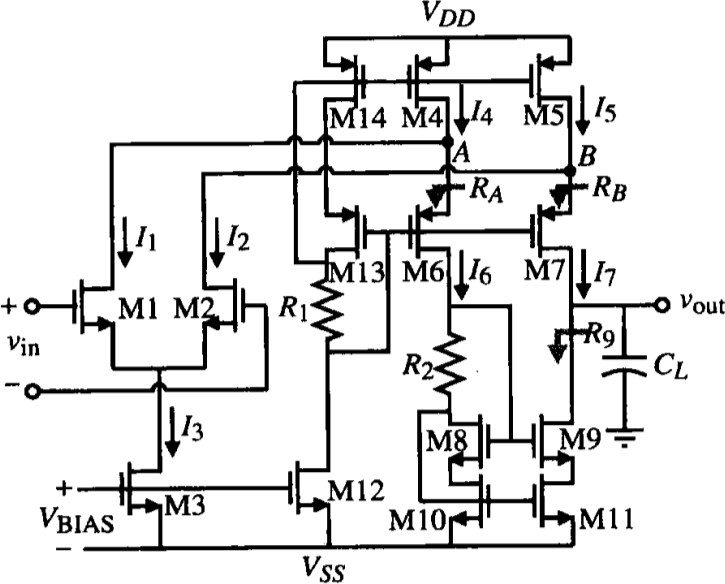


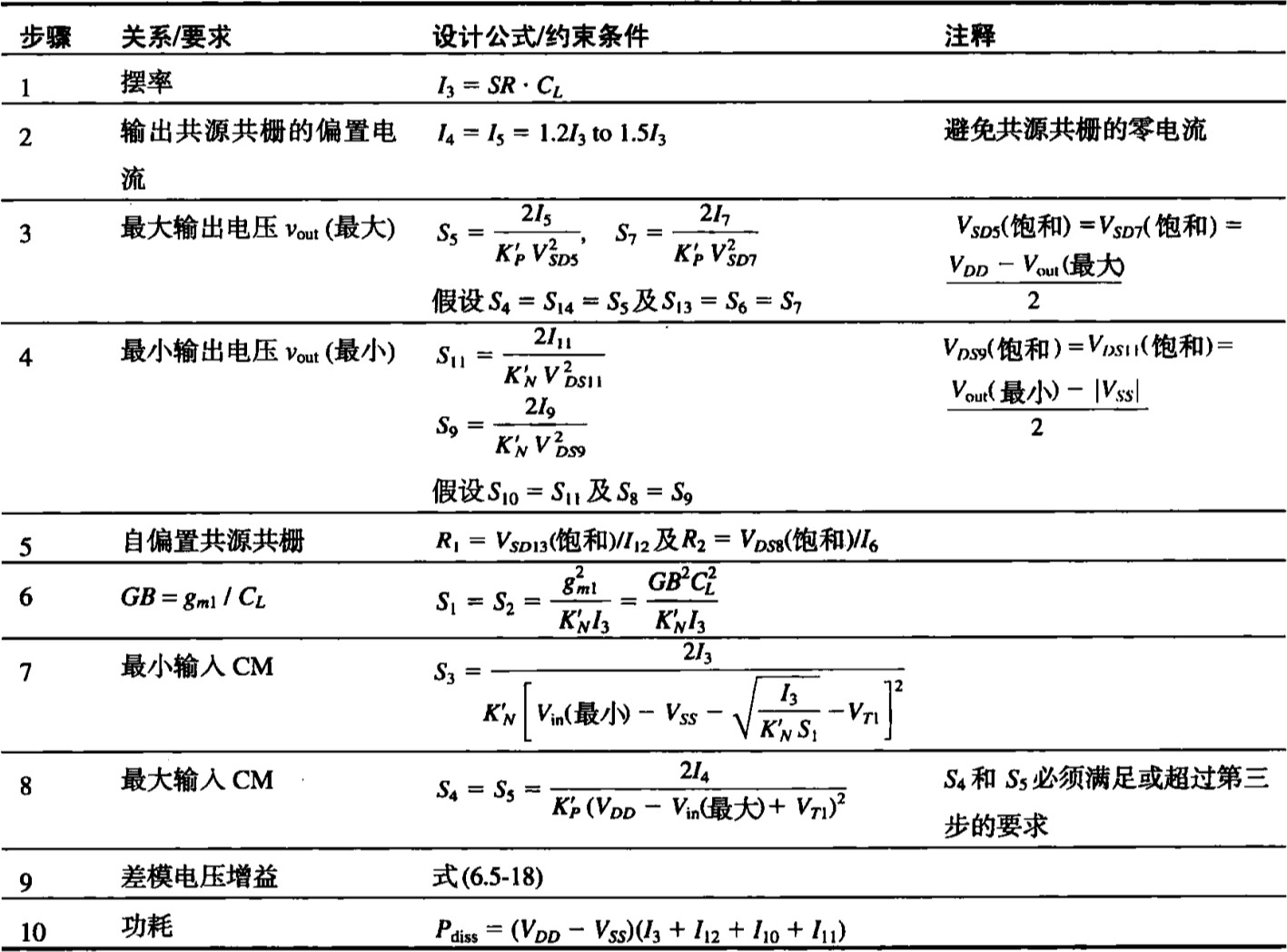




# 放⼤器理论计算



**图5 折叠式共源共栅放⼤器结构图（Allen中⽂版P244）**



**图6 折叠式共源共栅的设计⽅法（Allen中⽂版P247）**

根据Allen课本给出的电路结构和计算公式，具体的计算步骤如下：

1. **确定 ** **的值。**

 ，选择 ，则  。

1. **确定 的值。**



取   。



1. **确定 的值。**

其中 饱 和  ，



最 ⼤

得    ，

再由表中第8步可得，此时的 满⾜最⼩要求。



1. **确定 的值。**

按照书中例⼦中最坏情况计算，也就是电流全部流向 ，此时    ，

饱 和 ，

得    。

1. **确定 ** **。**

其中 饱 和 饱 和 ，



最 ⼩

同样为最坏情况，     ，

得     。

1. **确定 ** **的值。**

其中    ，

得   。

1. **确定 的值。**



已知  ，

得   。

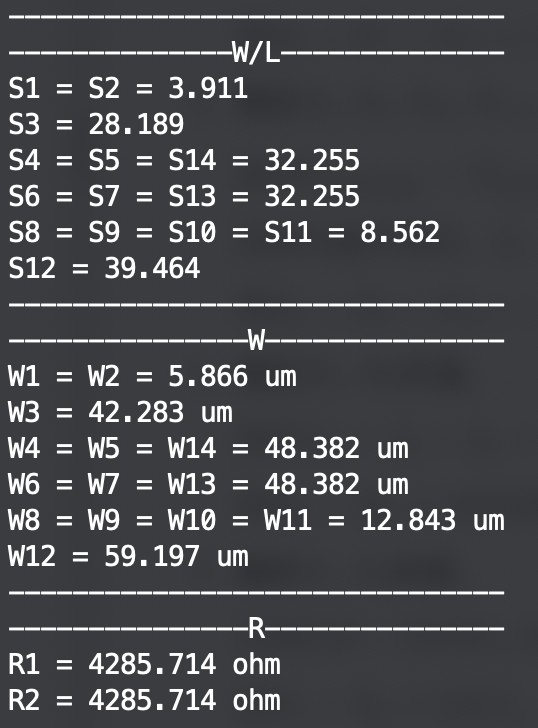
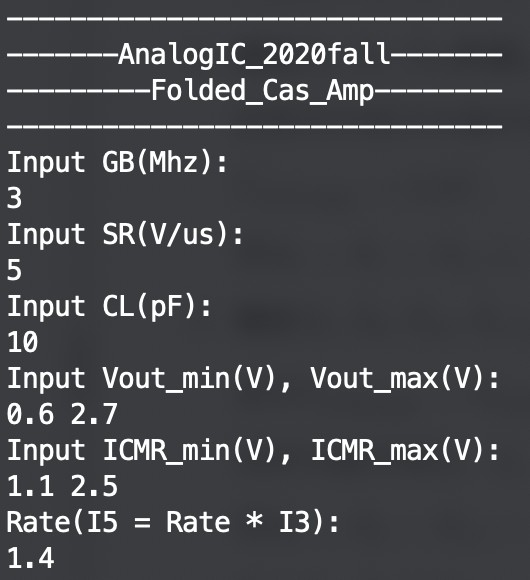
1. **确定 ** **的值。**

其中   ，

得    。

计算过程中所有的S代表栅宽栅⻓⽐W/L，在设计中去所有的 。

由于计算步骤较多，计算公式⽐较复杂，因此设计了⼀个简单的C程序，输⼊性能指标，输出参数的计算 结果，源码附在报告结尾。



**图7 计算程序输⼊输出**

初步理论设计参数为：

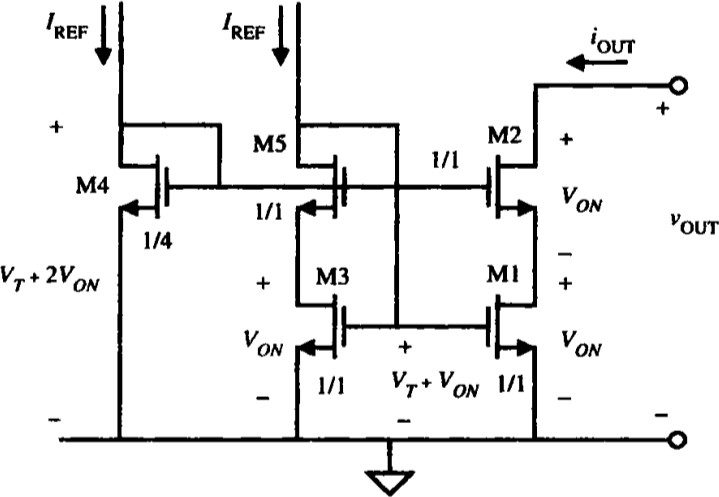
|  |  |  |  |
| --- | --- | --- | --- |
| **MOS管** | **W/L** | **W(um)** | **L(um)** |
| M1, M2 | 3.911 | 5.866(最终修改为8) | 1.5 |
| M3 | 28.189 | 42.283 | 1.5 |
| M4, M5, M14 | 32.255 | 48.382 | 1.5 |
| M6, M7, M13 | 32.255 | 48.382 | 1.5 |
| M8, M9, M10, M11 | 8.562 | 12.843 | 1.5 |
| M12 | 39.464 | 59.197 | 1.5 |

|  |  |
| --- | --- |
| **器件名称** | **数值** |
| R1, R2 | 4.28K ohm |
| CL | 10pF |

⼀开始增益不⾜3000，因此将W1和W2调⼤，修改为8 之后，可满⾜要求。

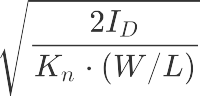
# 电流漏的设计及仿真

为了给  和  管提供稳定的偏置电压，需要设计⼀个电流漏，采⽤Allen书上P108的⾼摆幅共源共栅 电流漏结构。



**图8 ⾼摆幅共源共栅电流漏结构**

由于  ⼯作在饱和区，可以得到：

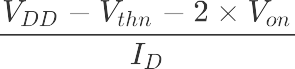


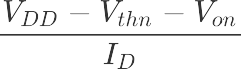
取     ，解得  。在此电路中，通过将 的栅宽栅

⻓⽐变为其他管⼦的 ，从⽽可以达到使 减⼩到最低值的⽬的。

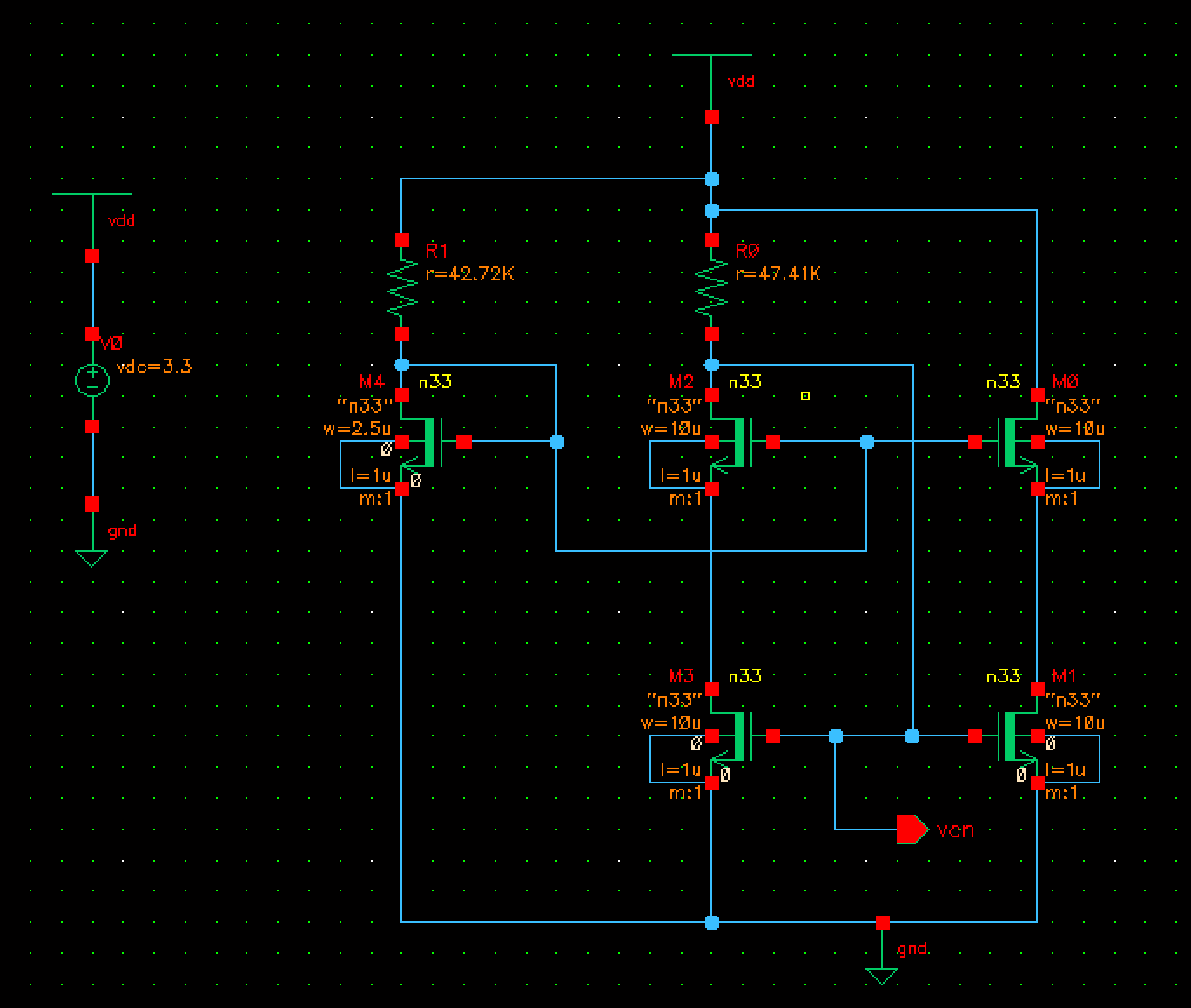


电阻⼤⼩为：

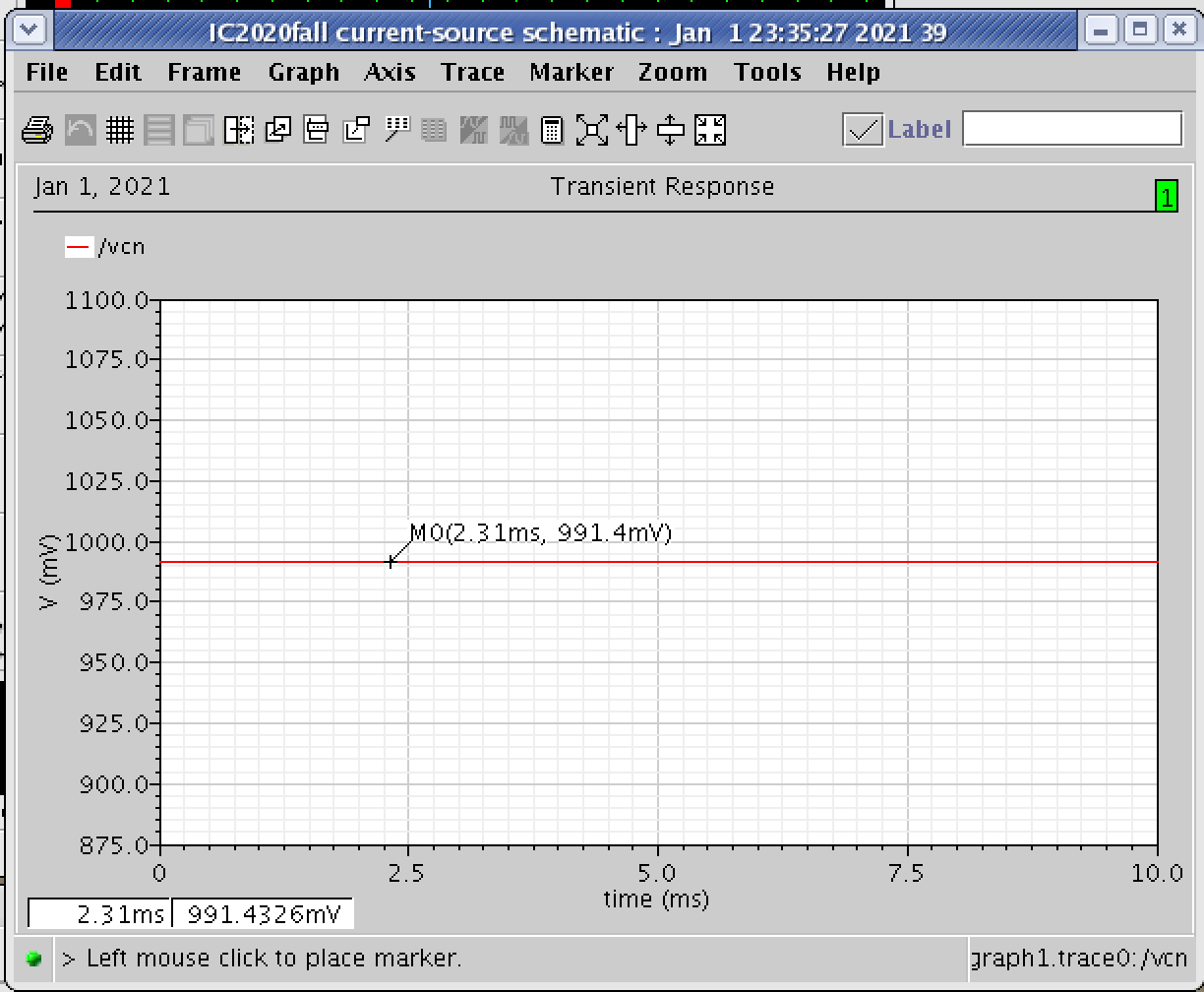


取  的栅极电压为输出电压，具体仿真电路为：

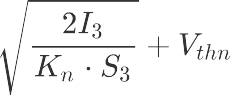


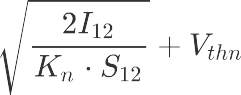
**图9 电流漏仿真原理图**

进⾏瞬态分析仿真，取仿真时间为10ms，仿真结果为：

**图10 电流漏瞬态分析结果**

理论上在之前的折叠共源共栅放⼤器中，  的偏置电压应该为：

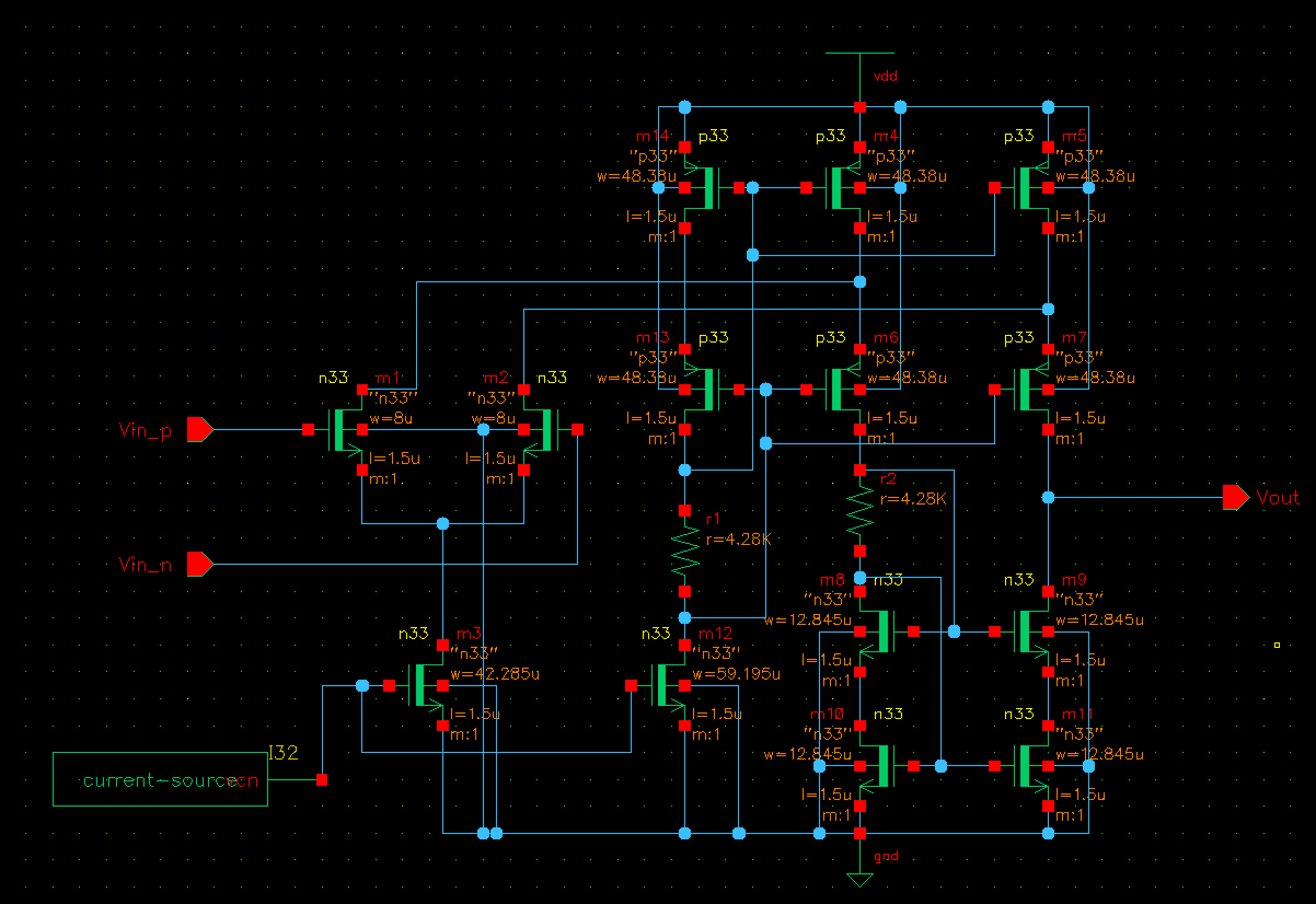


**因此，该电流漏能够满⾜设计要求。**

# 整体电路仿真以及性能测试

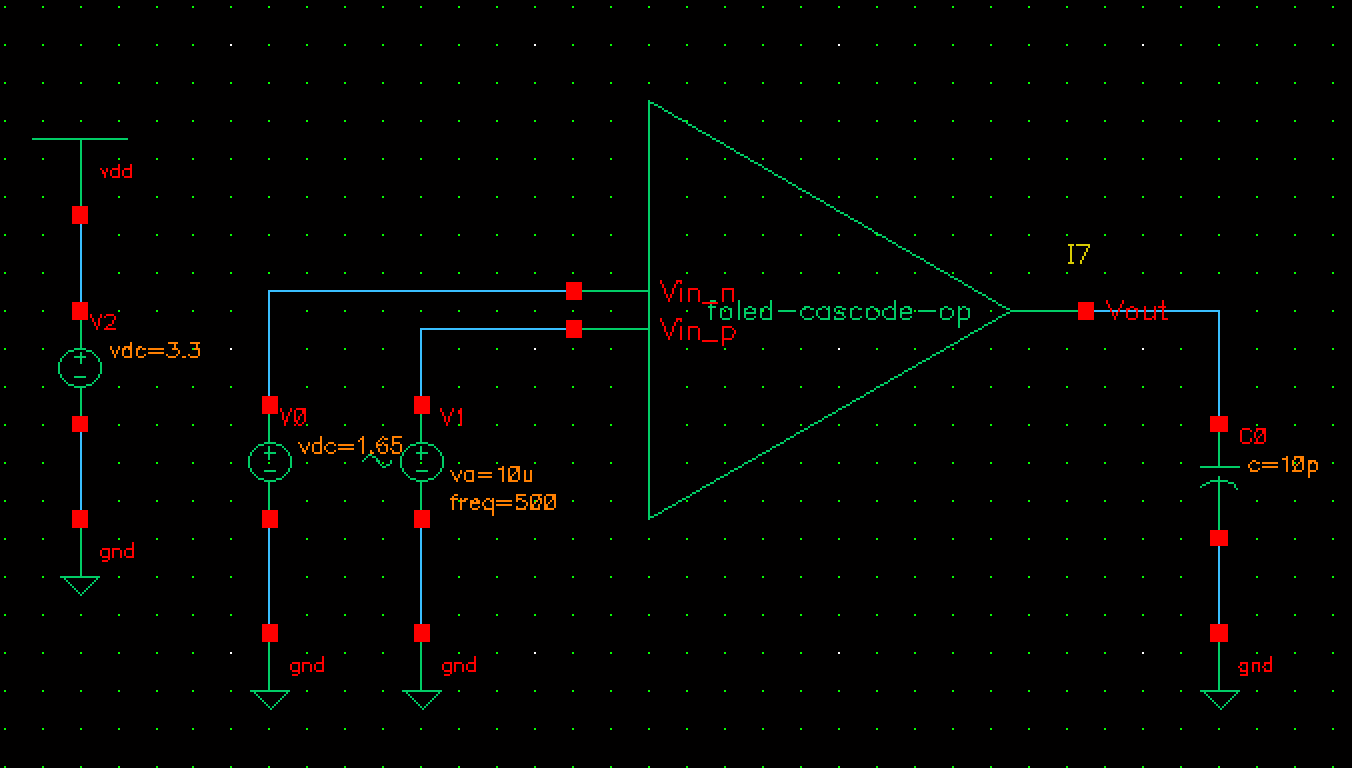
## 放⼤电路搭建



**图11 放⼤器原理图**

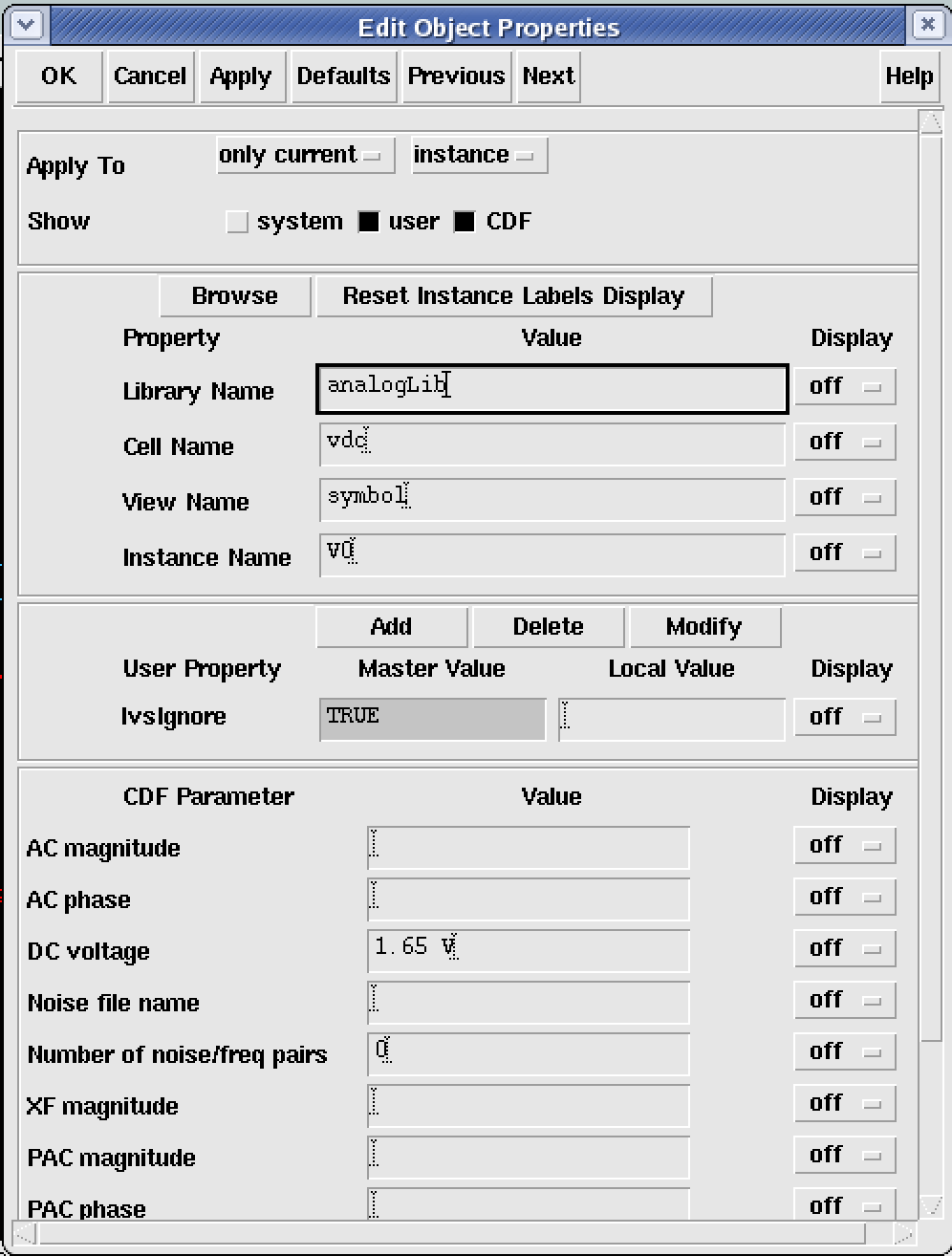
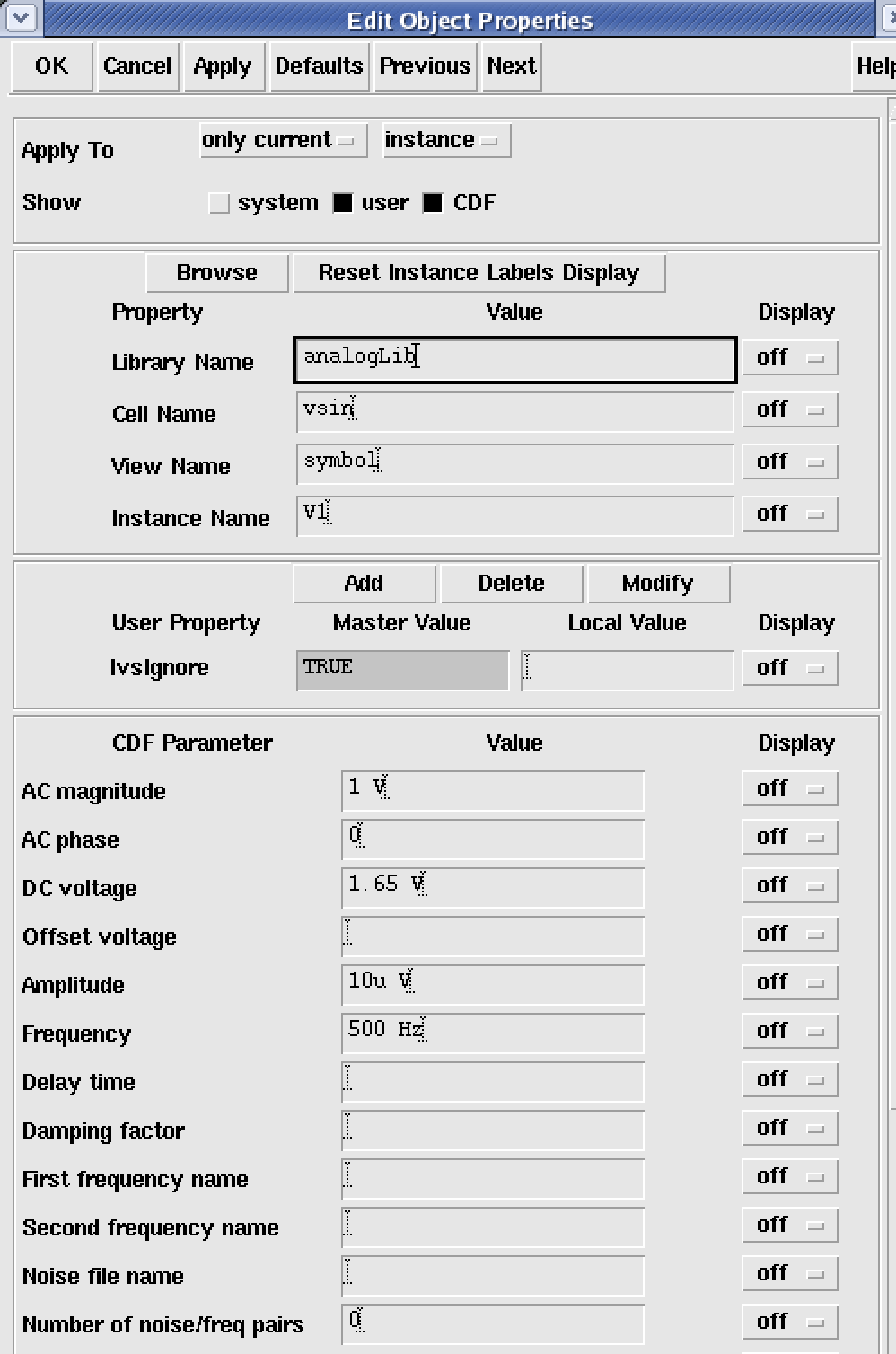
经过初步仿真，所有MOS管均⼯作在饱和区，电路设计基本正确。对其进⾏封装，进⾏各个性能指标的 测试。

## 开环电压增益Av测量



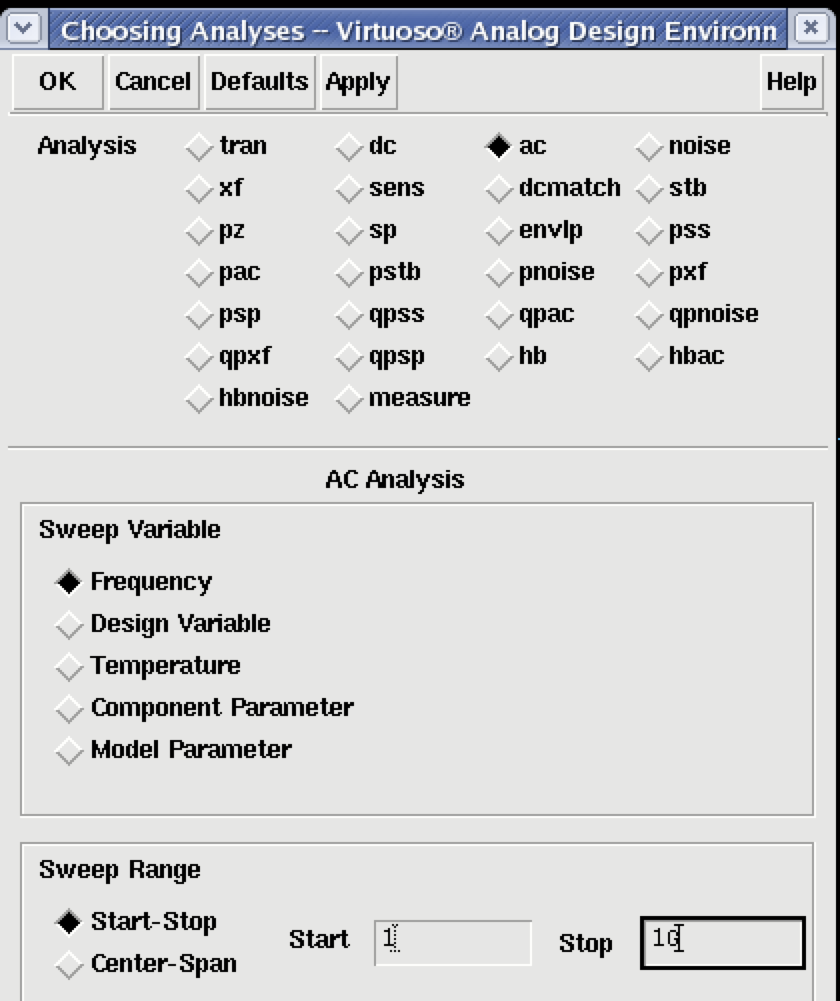
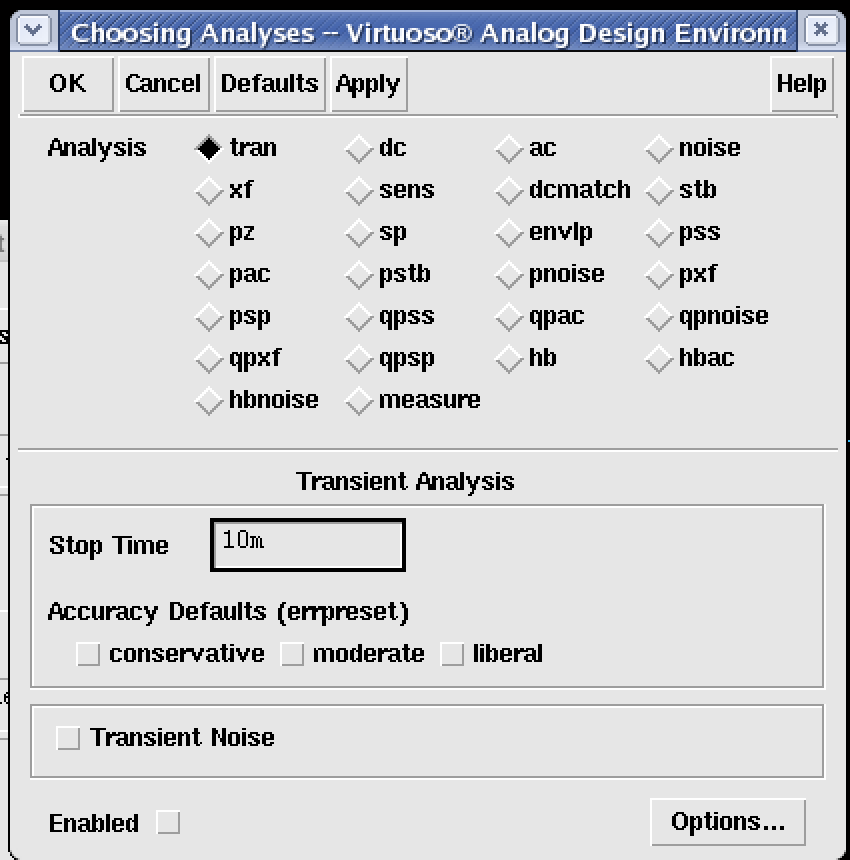
**图12 开环电压增益测量图**

放⼤器正极接⼀个差分输⼊幅值为10uV的500Hz正弦电源，负极接⼀个1.65V的直流电源，设置如下：



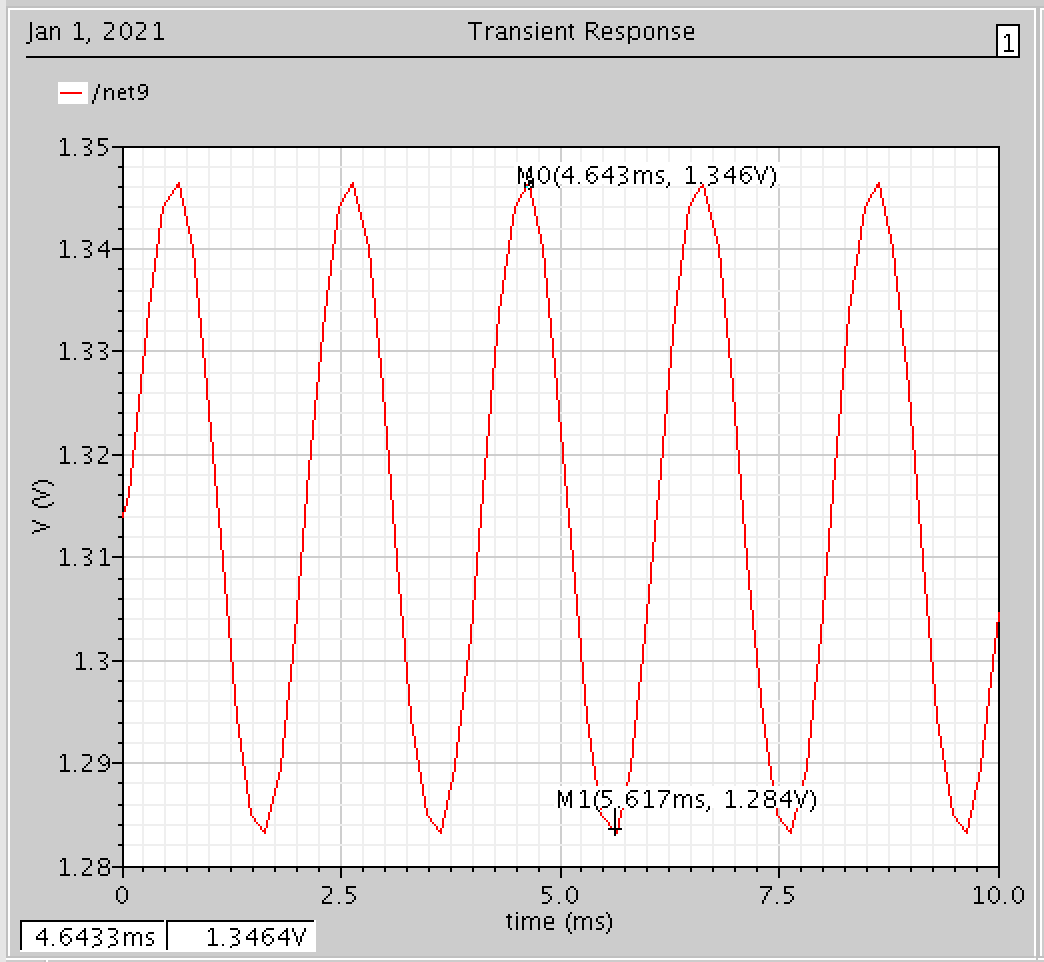
**图13 电压源设置**

进⾏仿真，选择瞬态分析，仿真时间设置为10m，AC分析频率从1Hz到1GHz，设置如下：



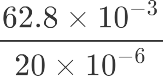
**图14 仿真设置**

选择输出端⼝，瞬态分析仿真结果如下：



**图15 瞬态分析结果**

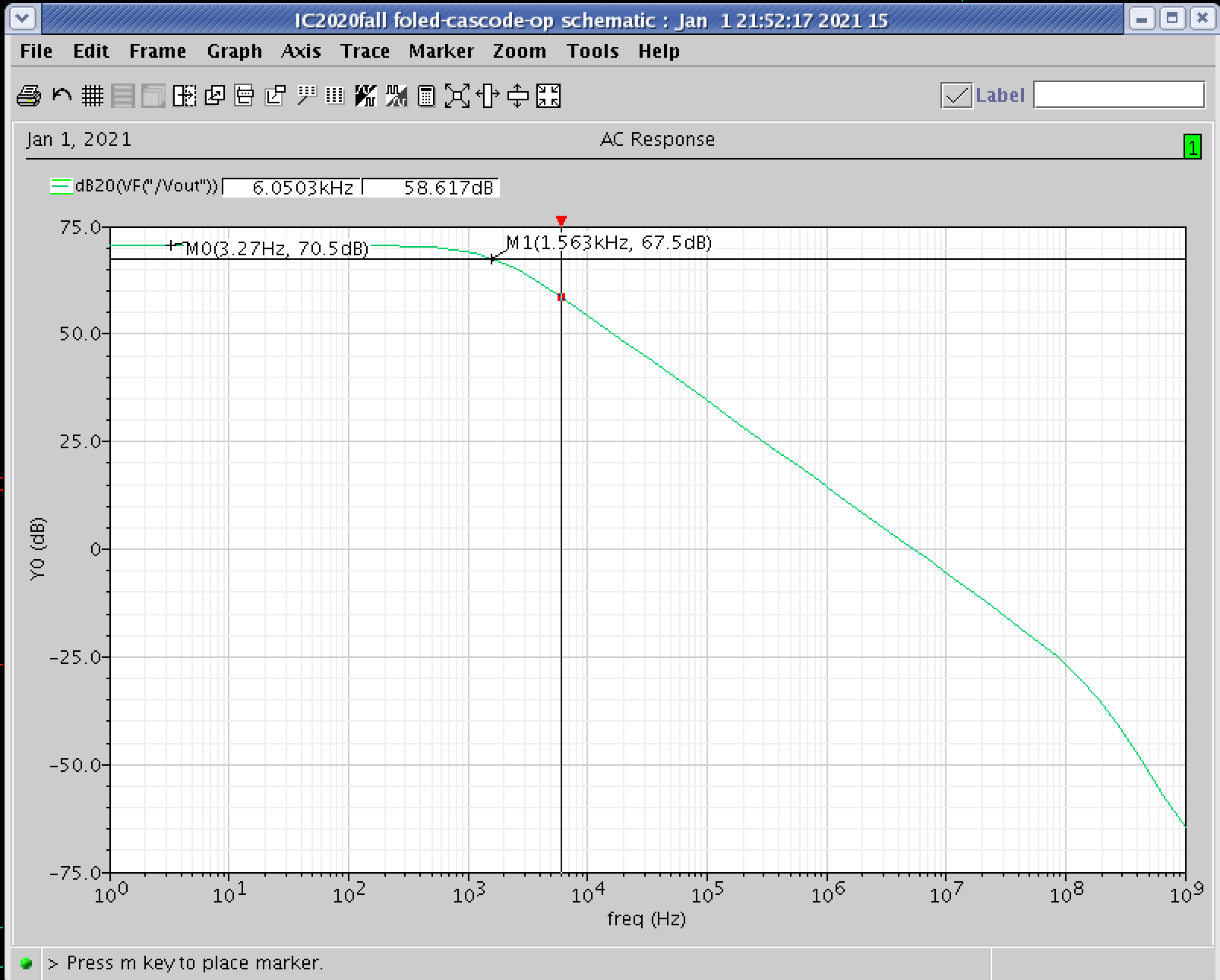
由图可得输出端⼝的峰峰值约为 ，则可以计算得到放⼤倍数为：





**增益⼤于3000，满⾜设计要求**。

## 带宽增益积GB测量

同样使⽤ 测试图和仿真设置，画出输出端⼝AC分析的db20曲线，选择Results->Direct Plot->db20。结果如下：

**图16 AC分析db20曲线**

由图可得，曲线的-3dB点处的频率为1.563KHz，可得带宽增益积：

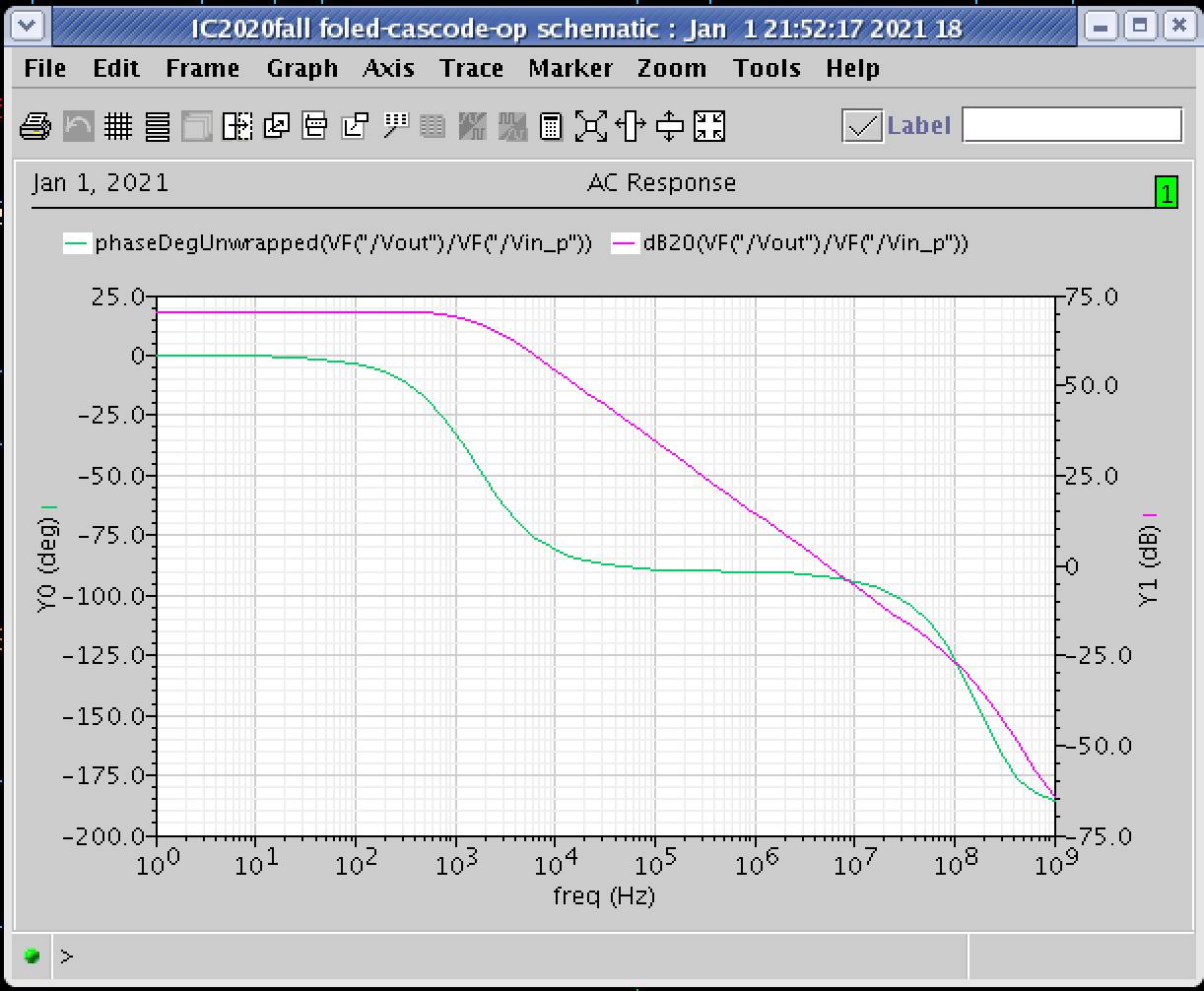




**带宽增益积⼤于3Mhz，满⾜设计要求。**

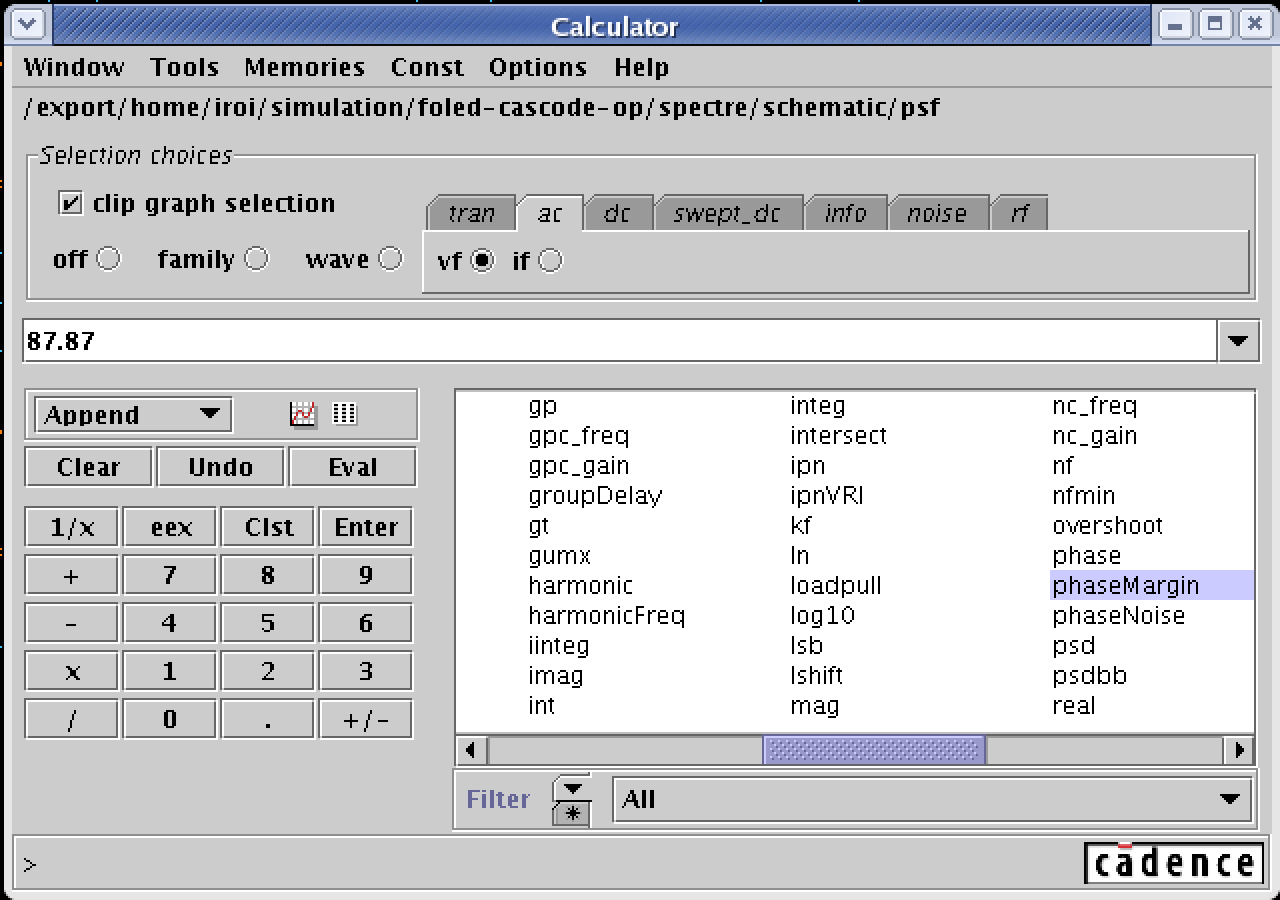
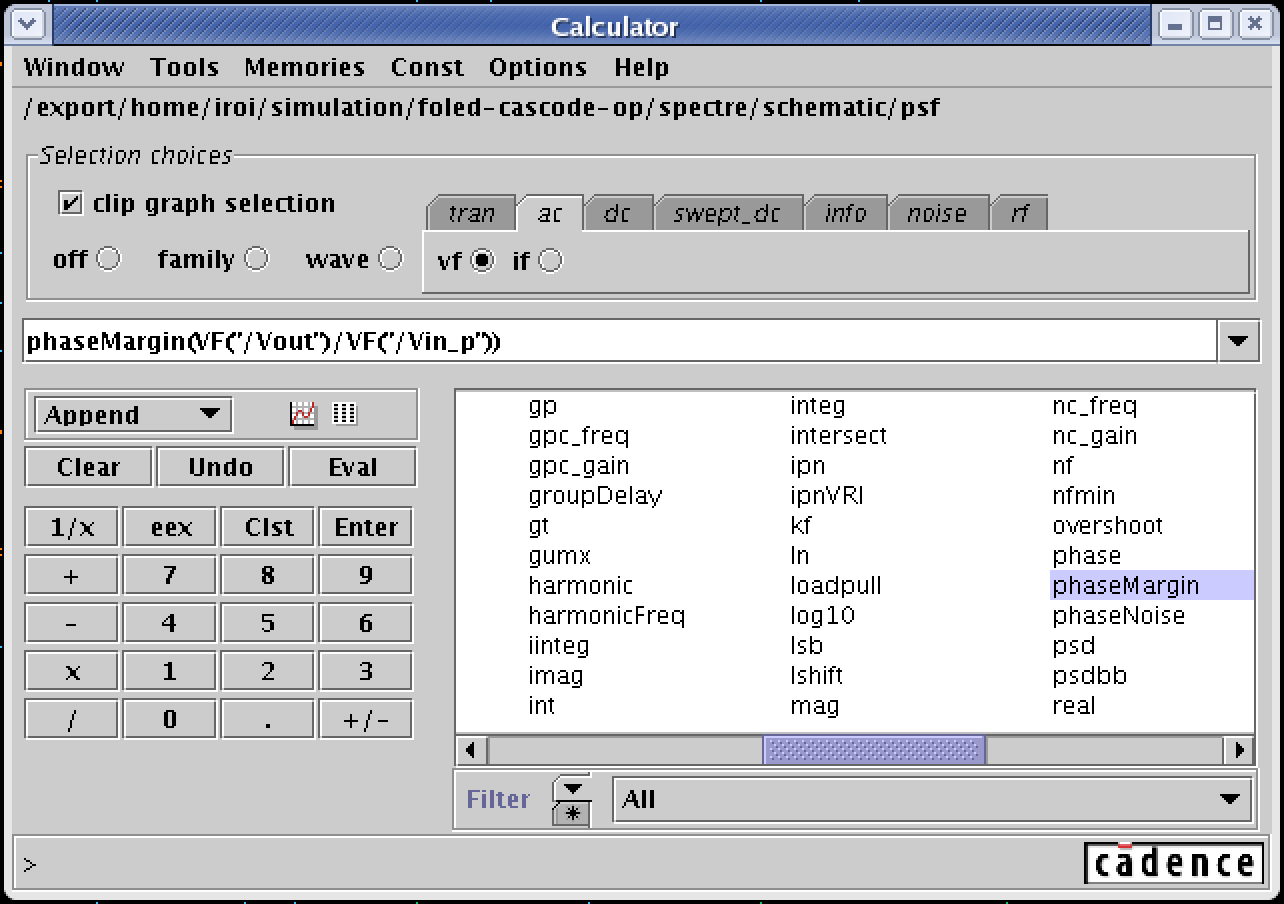
## 相位裕量Phase margin测量

同样使⽤ 测试图和仿真设置，画出AC Gain&Phase曲线，选择Results->Direct Plot->AC Gain&Phase，先点输出端⼝，在点正极输⼊端⼝。结果如下：



**图17 AC分析AC Gain & Phase曲线**

选择Tools->Calculator->ac->vf，先点击输出端⼝，再点击正极的输⼊端⼝，按下计算器上的/，最后找到phaseMargin计算，点击Eval得到结果，结果如下：

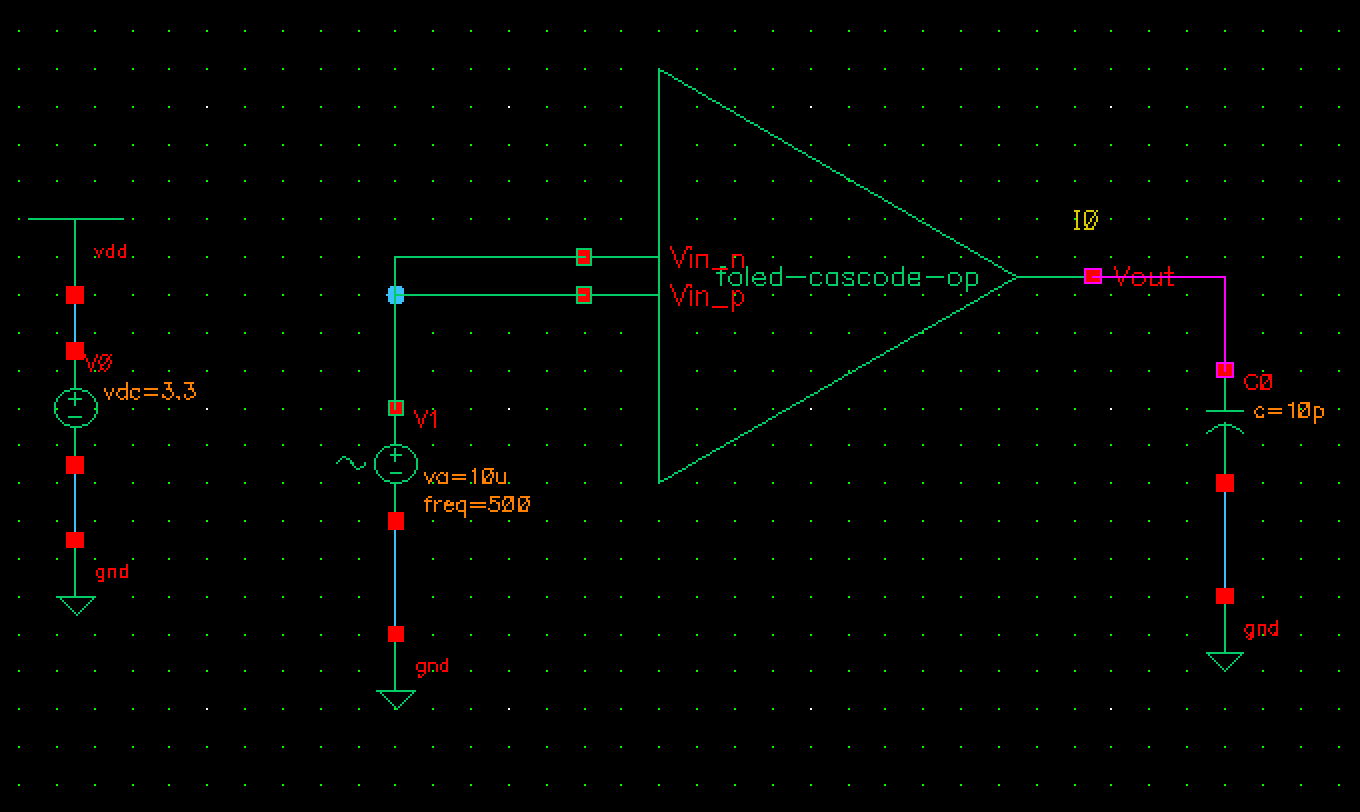


**图18 相位裕量计算**



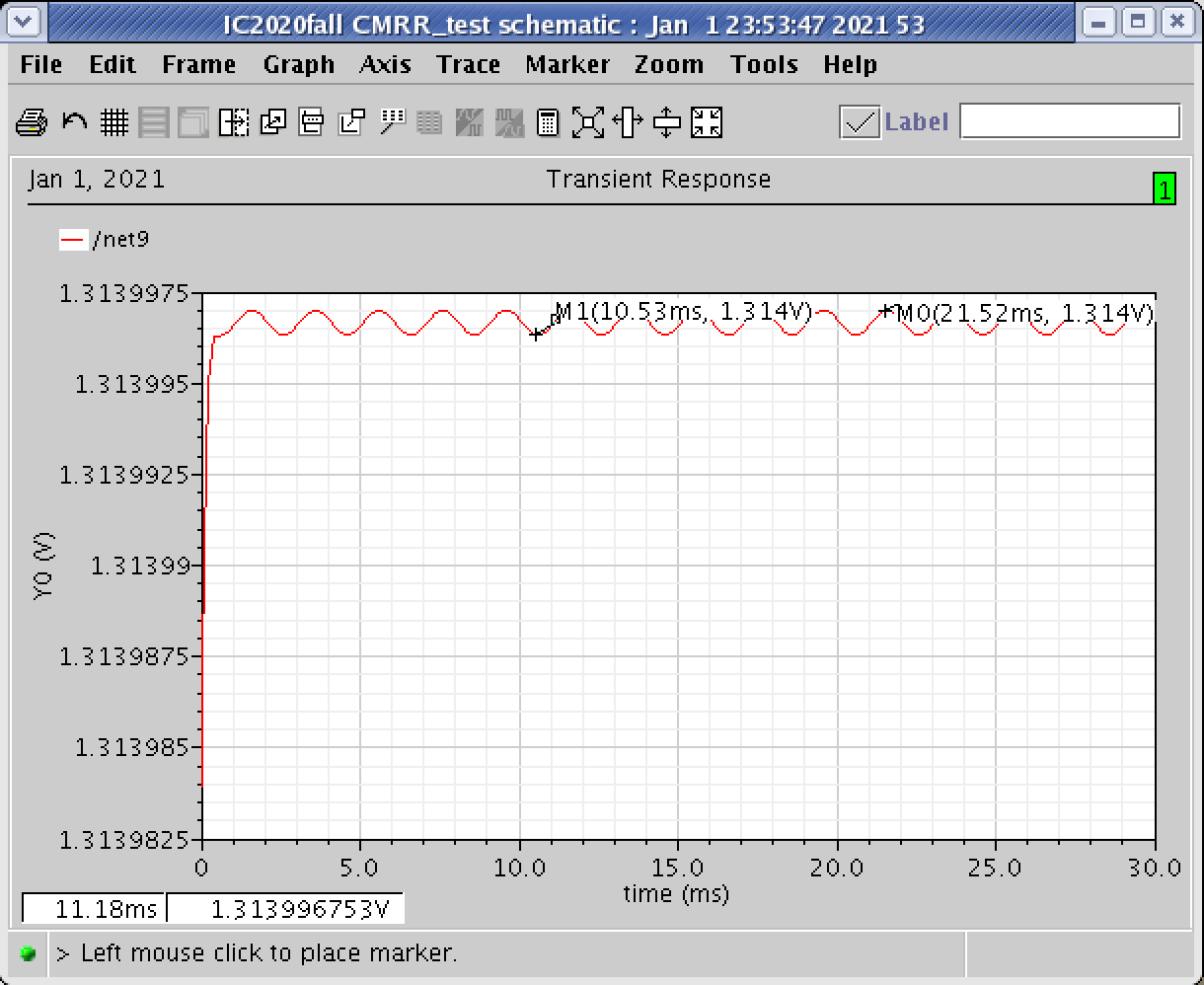
**相位裕量⼤于 ，满⾜设计要求。**

## 共模抑制⽐CMRR测量



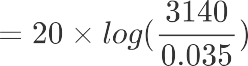
**图19 CMRR测量图**

将正极和负极共同接在⼀个正弦电压源上，设置与测量增益时的相同。对输出端做30ms瞬态分析，结果 如下：



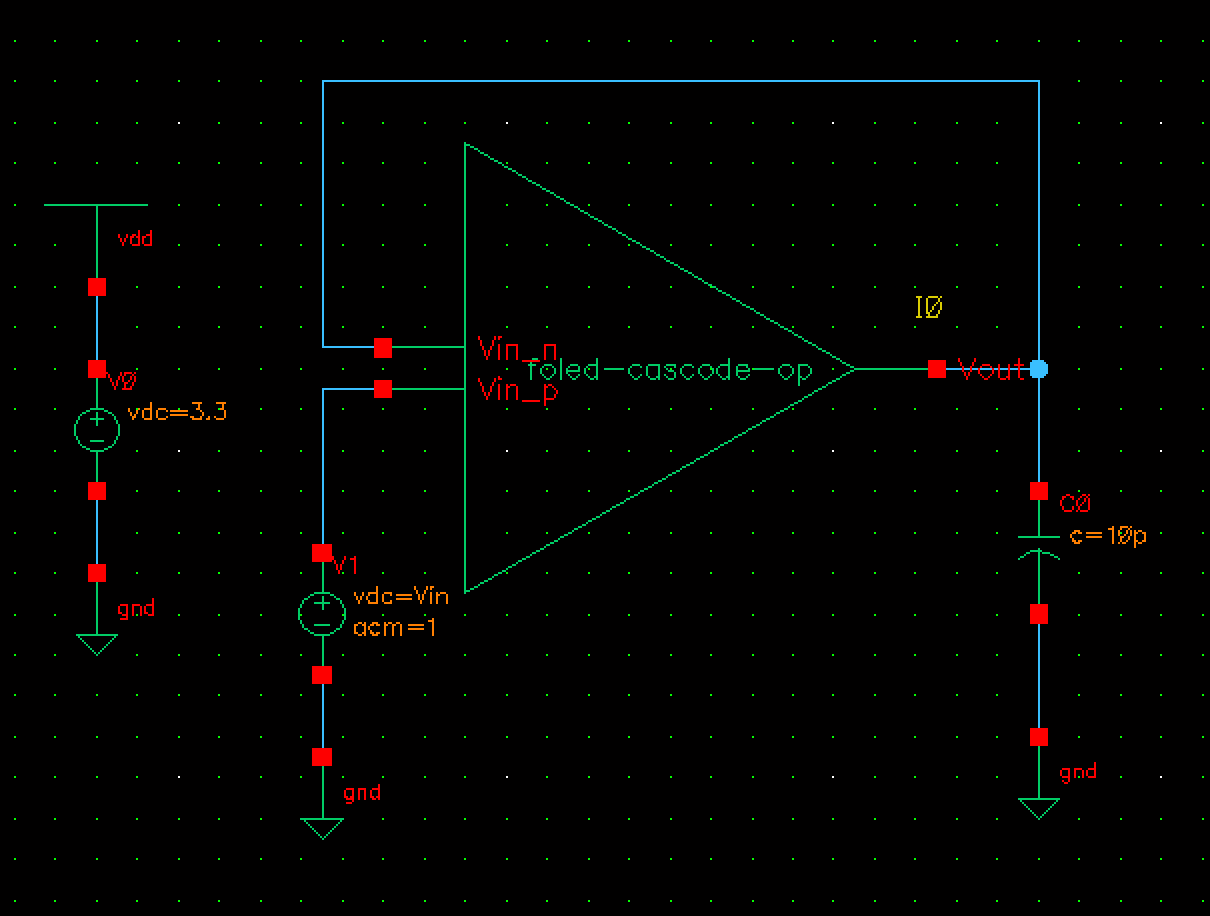
**图20 CMRR瞬态分析**

峰峰值约为 ，共模增益为 ，因此：



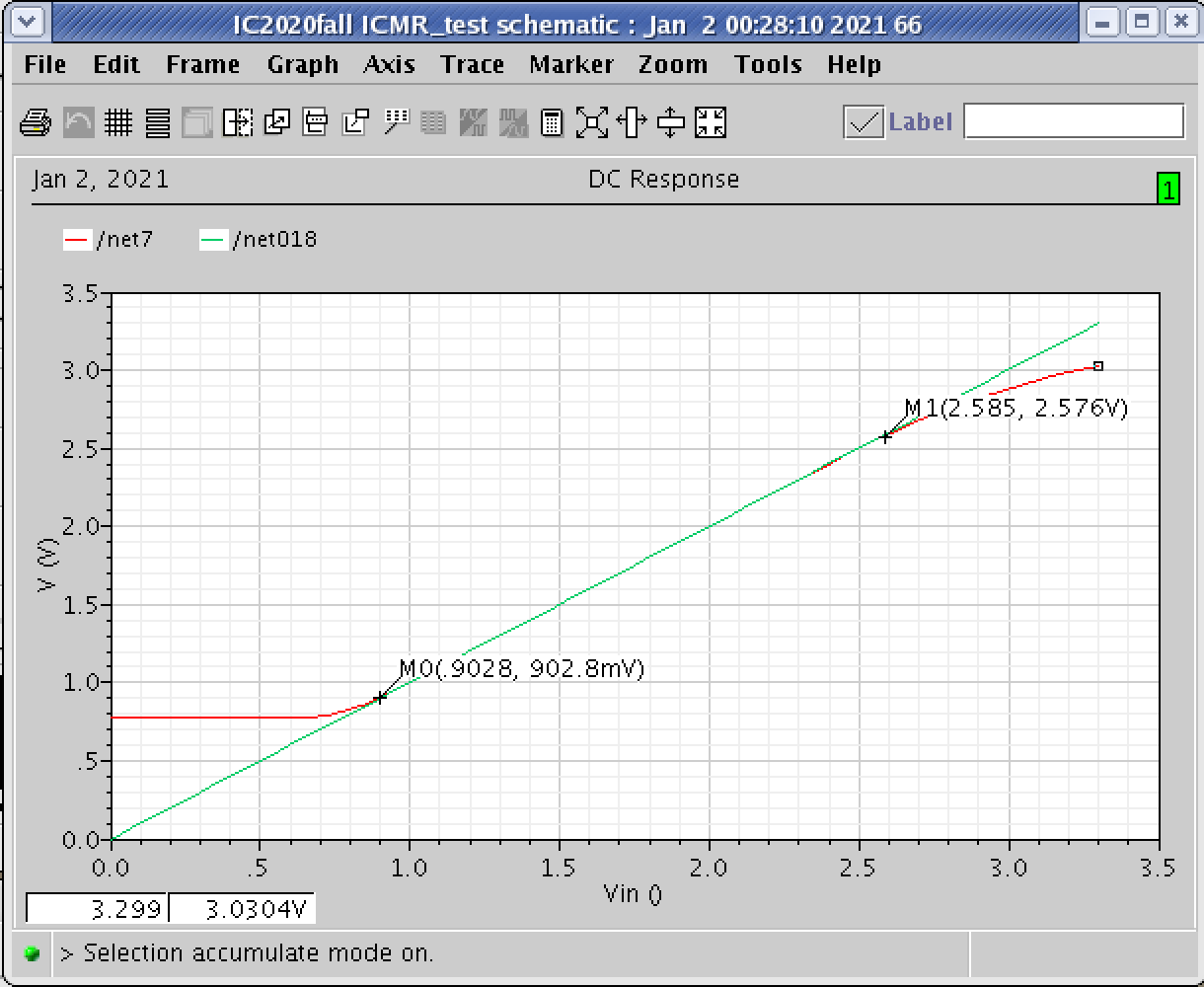
**说明该电路有不错的共模抑制效果。**

## 共模输⼊范围ICMR测量



**图21 ICMR测量图**

运放负极接到输出端，正极输⼊⼀个交流分量幅值1V，直流分量为变量Vin的电压源。选择输⼊和输出端

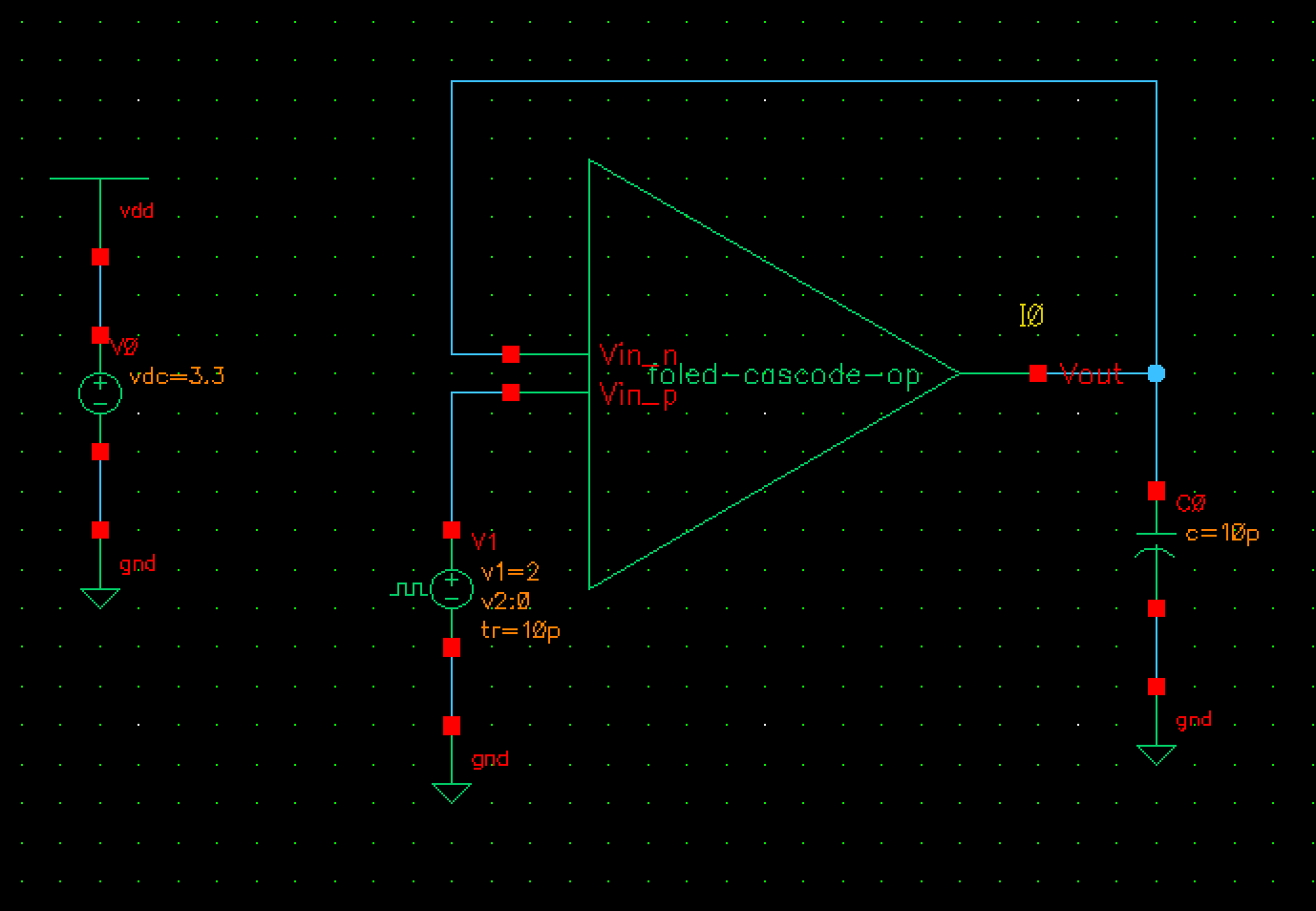
⼝，对变量Vin做DC分析，从0到3.3V扫描，结果如下：

**图22 ICMR DC分析**

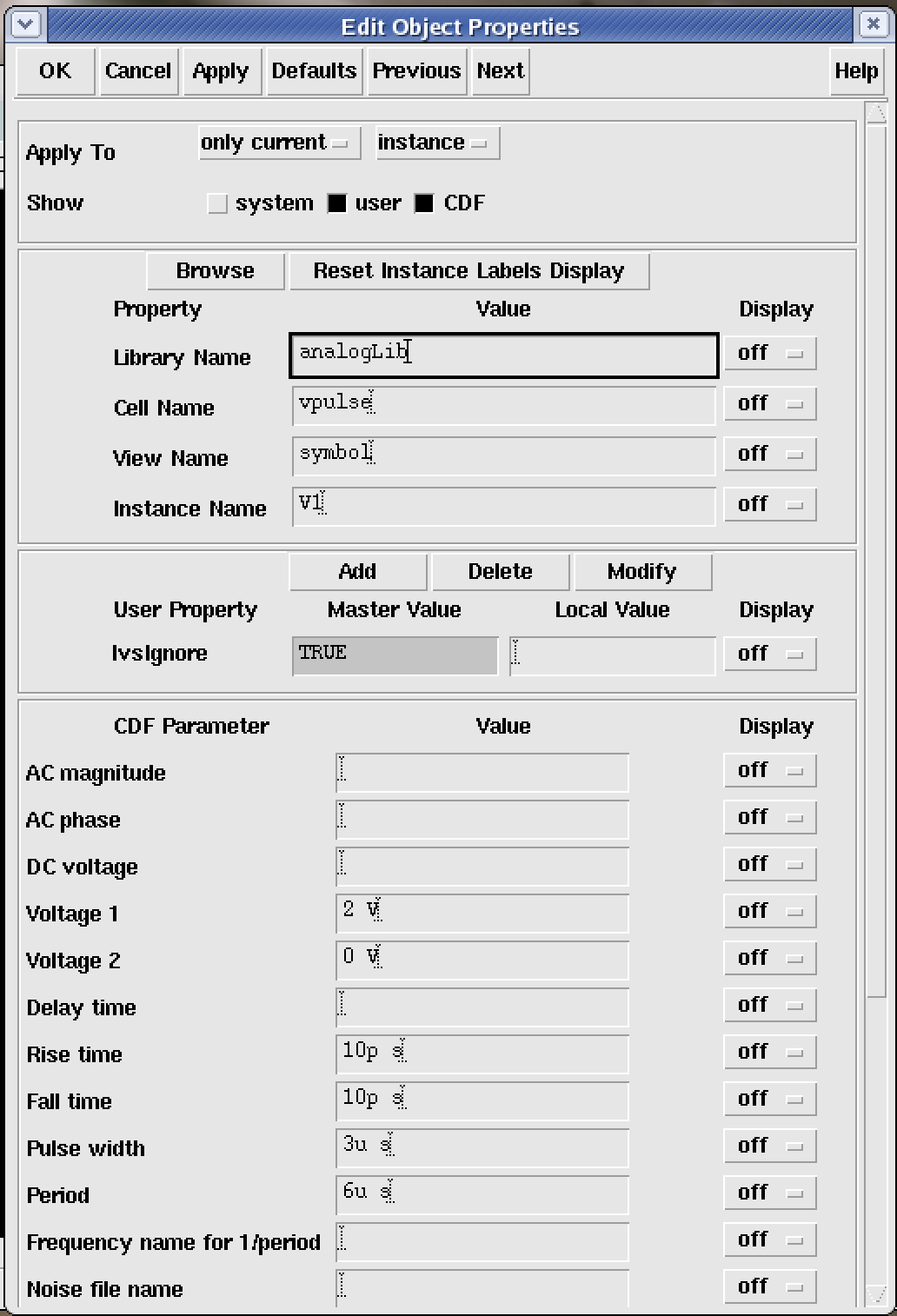
图中红线为输出电压曲线，绿线为输⼊电压曲线，可以看出两者在0.903V~2.576V的范围内保持重合。

**共模输⼊电压范围⼤于1.1V～2.5V，满⾜设计要求。**

## 压摆率SR测量

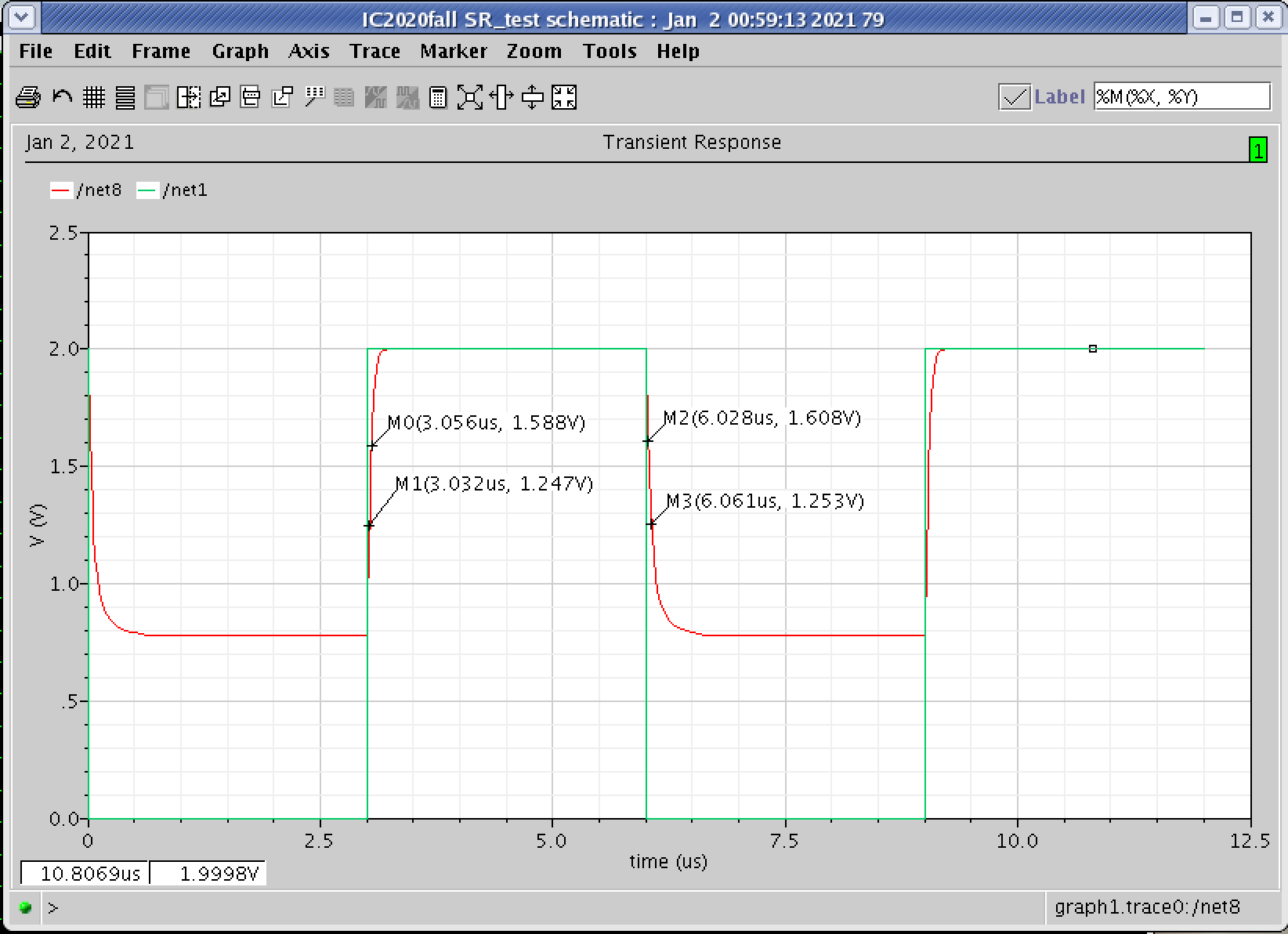


**图23 SR测量图**

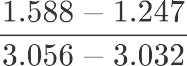
输⼊⼀个周期为6us，占空⽐为50%，⾼低电平分别为2V和0V的⽅波电压源，设置如下：

**图24 ⽅波电压源设置**

对输⼊输出端⼝进⾏瞬态分析，时常选择两个周期12us，结果如下：

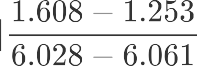


**图25 SR瞬态分析**

图中红⾊曲线为输出端⼝，绿⾊曲线为输⼊端⼝。 对于上升沿：

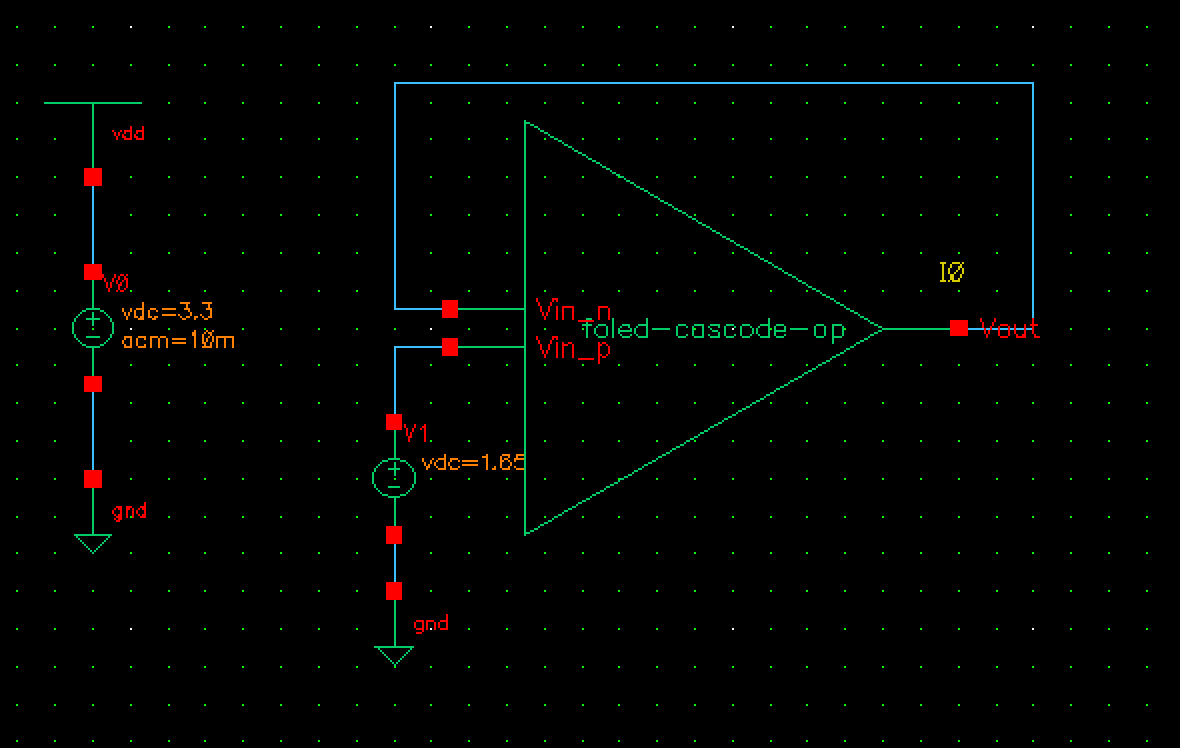
对于下降沿：





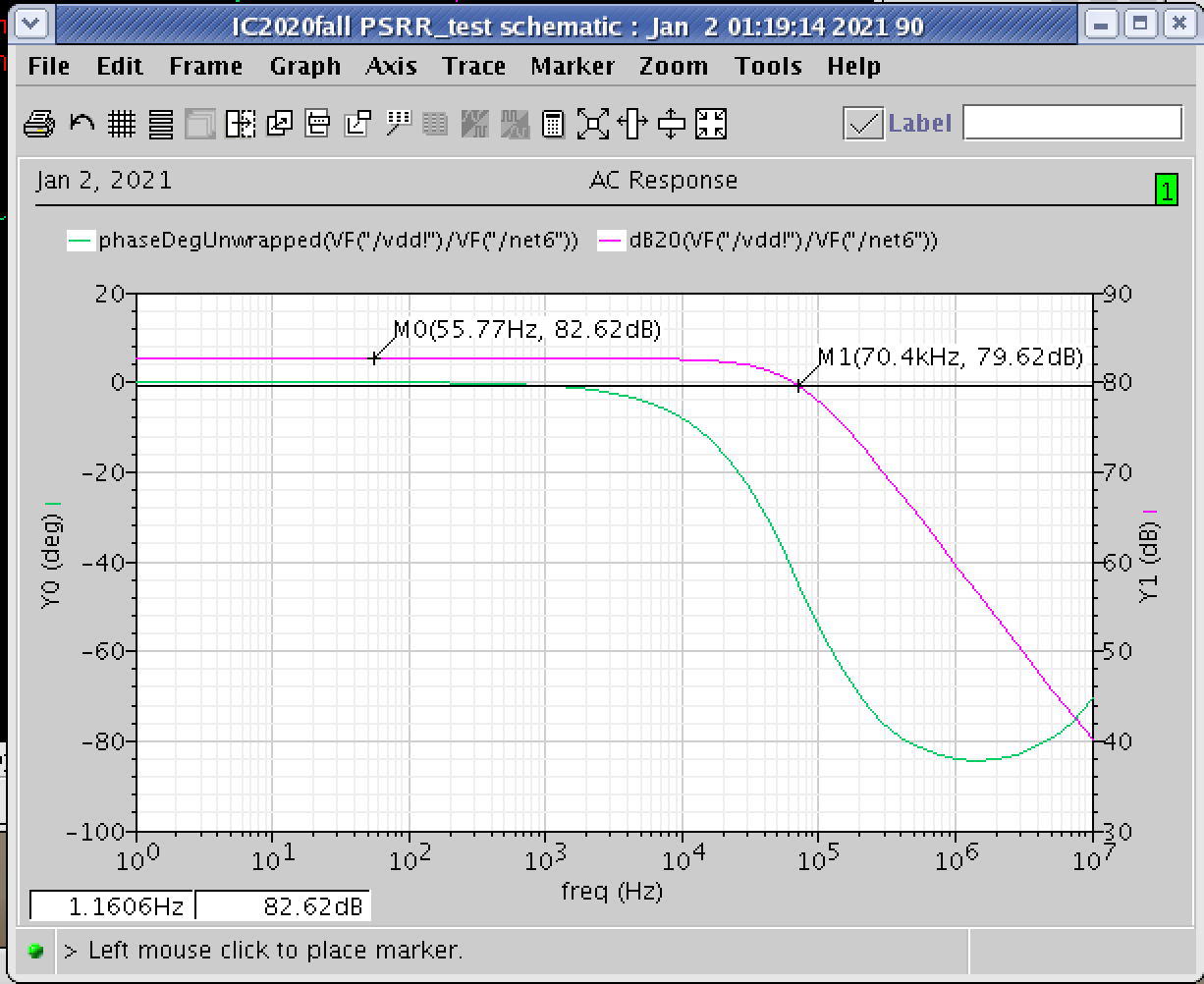
**压摆率SR⼤于5 ，满⾜设计要求。**

## 电源抑制⽐PSRR测试



**图26 PSRR测量图**

PSRR测量时不需要加电容负载，在Vdd上加⼀个幅值为10mV的交流分量。选择AC分析，频率从1Hz到10MHz，测量输出电压与交流⼩信号的⽐值，结果为：



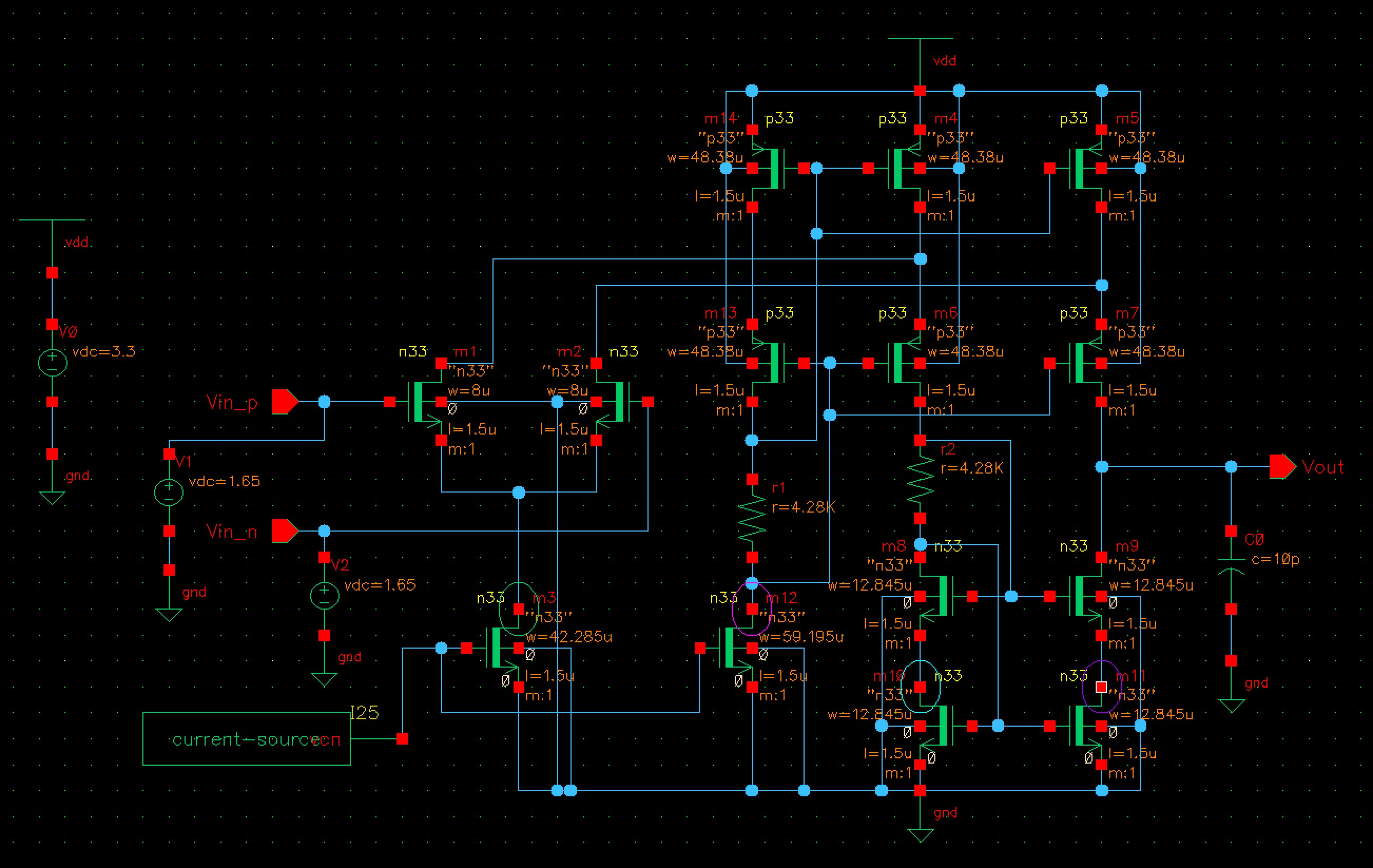
**图27 PSRR AC分析**

对于单电源供电电路：

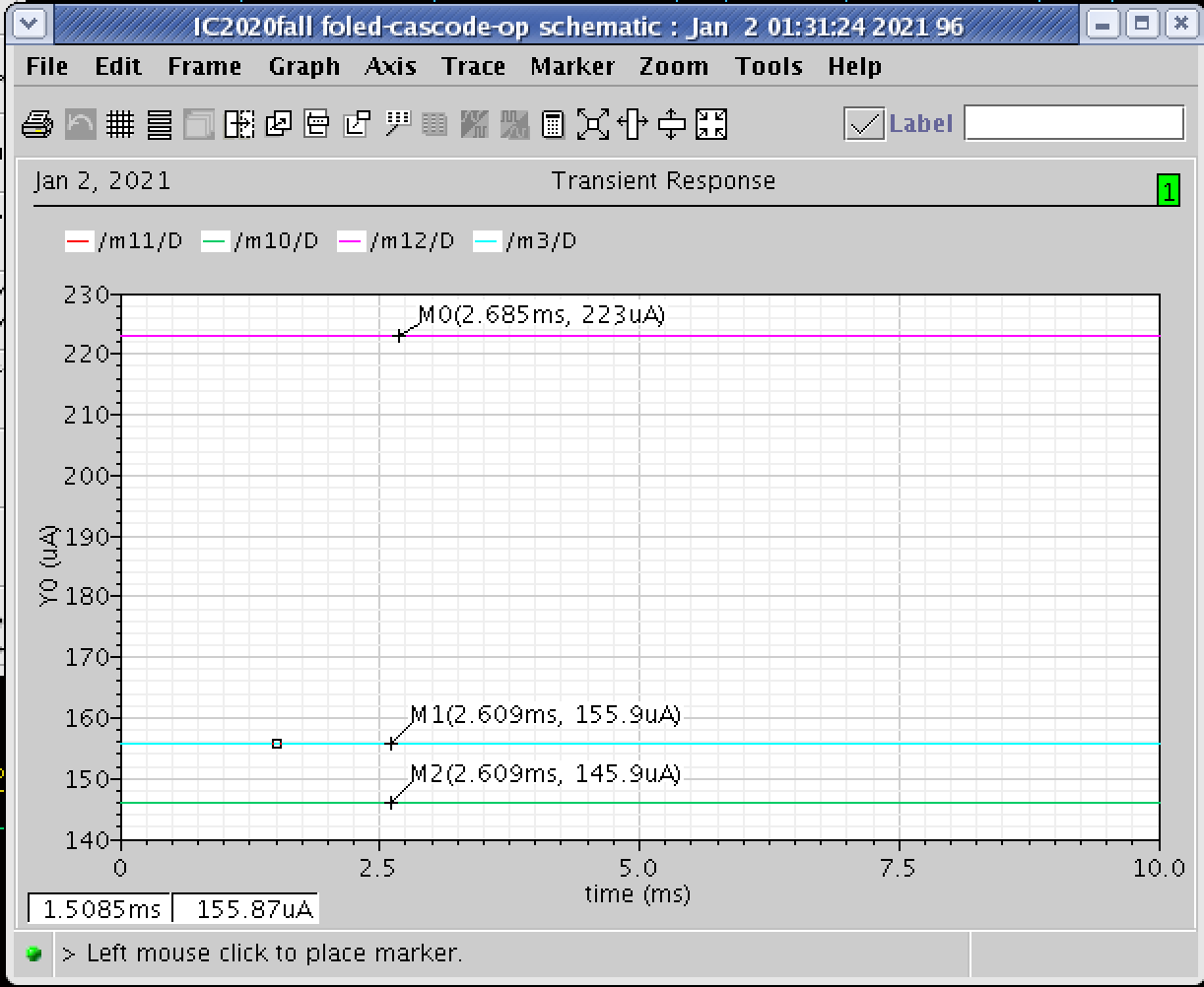


**从图中看出，PSRR在低频时约为82.62dB，-3dB点在70.4KHz处，说明该电路有良好的电源抑制效果。**

## Pdiss测量



**图28 Pdiss测量图**

正负极都接1.65V的直流电压源，⽤瞬态分析测得  ，结果如图：

**图29 Pdiss瞬态分析**





**Pdiss⼩于5mW，满⾜设计要求。**

# 实验总结

1. 通过本次实验，熟悉了Cadence仿真软件的使⽤，学会了设计折叠式共源共栅运算放⼤器，并能通 过仿真来测试性能参数，对模拟集成电路设计有了更深的理解。
2. 实验中有两个注意的点：
   1. 增益不够时，⾸先需要查看每个管⼦是否都⼯作在饱和状态，如都⼯作在饱和状态，可以尝试 调⼤ 。



* 1. 所有NMOS的体端都要接地，PMOS的体端接Vdd。

# 附: parameter.c 源代码

1. #include <stdio.h>
2. #include <math.h> 3

4 #define u 0.000001

5 #define m 0.001

6 #define M 1000000.0

7 #define K 1000.0

8

9 #define Kn 181.685 //402.675 //181.658 // 单 位 uA/(V^2)

10 #define Kp 48.227 //106.421 //48.227

1. #define Vtn 0.695
2. #define Vtp -0.672

13 #define lamN 0.04 //0.06

14 #define lamP 0.05 //0.08

1. #define Vdd 3.3
2. #define Vss 0.0
3. //#define n 1.25
4. #define PI 3.1415
5. #define L 1.5 20
6. double GB, SR, CL;
7. double Vout\_min, Vout\_max, ICMR\_min, ICMR\_max;
8. double Av, Pdiss;
9. double n; 25

26 int main() { 27

1. printf(" \n");
2. printf("-------AnalogIC\_2020fall \n");
3. printf("---------Folded\_Cas\_Amp \n");
4. printf(" \n");
5. printf("Input GB(Mhz): \n");
6. scanf("%lf", &GB);
7. printf("Input SR(V/us): \n");
8. scanf("%lf", &SR);
9. printf("Input CL(pF): \n");
10. scanf("%lf", &CL);

|  |  |
| --- | --- |
| 38 | printf("Input Vout\_min(V), Vout\_max(V): \n"); |
| 39 | scanf("%lf %lf", &Vout\_min, &Vout\_max); |
| 40 | printf("Input ICMR\_min(V), ICMR\_max(V): \n"); |
| 41 | scanf("%lf %lf", &ICMR\_min, &ICMR\_max); |
| 42 | printf("Rate(I5 = Rate \* I3): \n"); |
| 43 | scanf("%lf", &n); |
| 44 |  |
| 45 | double I1, I3, I5, I7, Vsd5, Vsd9; |
| 46 | I3 = SR \* CL; |
| 47 | I5 = n \* I3; |
| 48 | I7 = I5 - (I3 / 2); |
| 49 | I1 = I3 / 2; |
| 50 | printf(" \n"); |
| 51 | printf("------------Current \n"); |
| 52 | printf("I3 = %.3lf uA\n", I3); |
| 53 | printf("I1 = I2 = %.3lf uA\n", I1); |
| 54 | printf("I4 = I5 = %.3lf uA\n", I5); |
| 55 | printf("I6 = I7 = ... = I11 = %.3lf uA\n", I7); |
| 56 |  |
| 57 | Vsd5 = 0.5 \* (Vdd - Vout\_max); |
| 58 | Vsd9 = 0.5 \* (Vout\_min - Vss); |
| 59 |  |
| 60 | double S1, S3, S5, S7, S9, S5\_min, V3, V5, S12; |
| 61 | double W1, W3, W5, W7, W9, W12; |
| 62 | S5 = (2 \* I5) / (Kp \* Vsd5 \* Vsd5); |
| 63 | S7 = (2 \* I5) / (Kp \* Vsd5 \* Vsd5); |
| 64 | S9 = (2 \* I5) / (Kn \* Vsd9 \* Vsd9); |
| 65 | S1 = (4 \* PI \* PI \* GB \* GB \* CL \* CL) / (Kn \* I3); |
| 66 | V3 = ICMR\_min - Vss - sqrt(I3 / (Kn \* S1)) - Vtn; |
| 67 | S3 = (2 \* I3) / (Kn \* V3 \* V3); |
| 68 | V5 = Vdd - ICMR\_min - Vtp; |
| 69 | S5\_min = (2 \* I5) / (Kp \* V5 \* V5); |
| 70 | if (S5\_min >= S5) S5 = S5\_min; |
| 71 | S12 = S3 \* n; |
| 72 | W1 = L \* S1; |
| 73 | W3 = L \* S3; |
| 74 | W5 = L \* S5; |
| 75 | W7 = L \* S7; |
| 76 | W9 = L \* S9; |
| 77 | W12 = L \* S12; |
| 78 | printf(" \n"); |
| 79 | printf("--------------W/L \n"); |
| 80 | printf("S1 = S2 = %.3lf\n", S1); |
| 81 | printf("S3 = %.3lf\n", S3); |
| 82 | printf("S4 = S5 = S14 = %.3lf\n", S5); |
| 83 | printf("S6 = S7 = S13 = %.3lf\n", S7); |
| 84 | printf("S8 = S9 = S10 = S11 = %.3lf\n", S9); |
| 85 | printf("S12 = %.3lf\n", S12); |
| 86 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 87 | printf(" \n"); | | |
| 88 | printf("---------------W \n"); | | |
| 89 | printf("W1 = W2 = %.3lf um\n", W1); | | |
| 90 | printf("W3 = %.3lf um\n", W3); | | |
| 91 | printf("W4 = W5 = W14 = %.3lf um\n", W5); | | |
| 92 | printf("W6 = W7 = W13 = %.3lf um\n", W7); | | |
| 93 | printf("W8 = W9 = W10 = W11 = %.3lf um\n", W9); | | |
| 94 |  | printf("W12 = %.3lf um\n", W12); |  |
| 95 |  |  |  |
| 96 |  |  |  |
| 97 |  | double R1, R2; |  |
| 98 |  | R1 = Vsd5 / (I5 \* u); |  |
| 99 |  | R2 = Vsd5 / (I5 \* u); |  |
| 100 |  | printf(" \n"); |  |
| 101 |  | printf("---------------R \n"); |  |
| 102 |  | printf("R1 = %.3lf ohm\n", R1); |  |
| 103 |  | printf("R2 = %.3lf ohm\n", R2); |  |
| 104 |  |  |  |
| 105 |  | double gm1, gm3, gm5, gm7, gm9, gds1, gds3, gds5, | gds7, gds9; |
| 106 |  | gm1 = sqrt(I3 \* Kn \* S1); gds1 = (I3 / 2) \* lamN; |  |
| 107 |  | gm3 = sqrt(2 \* I3 \* Kn \* S3); gds3 = I3 \* lamN; |  |
| 108 |  | gm5 = sqrt(2 \* I5 \* Kp \* S5); gds5 = I5 \* lamP; |  |
| 109 |  | gm7 = sqrt(2 \* (I5 - I3/2) \* Kp \* S7); gds7 = (I5 | - I3/2) \* lamP; |
| 110 |  | gm9 = sqrt(2 \* (I5 - I3/2) \* Kn \* S9); gds9 = (I5 | - I3/2) \* lamN; |
| 111 |  |  |  |
| 112 |  | double Rii, Riii, Rout, k; |  |
| 113 |  | Rii = gm9 \* (1 / (gds9 \* gds9)); |  |
| 114 |  | Riii = gm7 \* (1 / gds7) \* (1 / (gds1 + gds5)); |  |
| 115 |  | Rout = (Rii \* Riii) / (Rii + Riii); |  |
| 116 |  | k = Rout \* (gds1 + gds5) \* gds7 / gm7; |  |
| 117 |  | Av = (2 + k) \* gm1 \* Rout / (2 + 2 \* k); |  |
| 118 |  | Pdiss = (Vdd - Vss) \* (I3 + 3 \* I5) \* m; |  |
| 119 |  |  |  |
| 120 |  | printf(" \n"); |  |
| 121 |  | printf("-----------Av, Pdiss \n"); |  |
| 122 |  | printf("Av = %.3lf\n", Av); |  |
| 123 |  | printf("Pdiss = %.3lf mW\n", Pdiss); |  |
| 124 |  |  |  |
| 125 | } |  |  |
| 126 |  |  |  |