Design of an Operational Transconductance Amplifier

tp24222
Thomas Plantin *(pledged)*

EE382M
Fall 2018

Course Instructor: Dr. Deji Akinwande
Teaching Assistants: Ajit Gopalakrishnan, Wei Shi

Parameter	**Specification for EE338L**	**Specifications Achieved**
Technology	0.18µm process[1], N-well, nominal corner	✓
Supply Voltage	1.8V (no scaling allowed)	1.8 V
Temperature	25°C	25°C
Power Dissipation	**Minimize**	672.66 µW
C_L	≥ 2pF	✓
Closed-loop gain	2	1.9981
C_s	$C_s \leq C_L$	✓
Dynamic Range[2]	70 dB	75.78 dB
Static settling error	≤0.05%	0.0498%
Dynamic settling error	≤0.05%	0.0017%
Setting time[3]	35ns	32.97 ns
CMRR[4] at DC	>100dB	129.85 dB
PSRR[4] +/- at DC (10mV differential offset)	>80dB	PSRR+ = 132.40 dB; PSRR- = 131.95 dB
Passives	Total ideal capacitance < 10pF; Total ideal resistance < 100KΩ	✓
OTA input stage	NMOS or PMOS differential pair with a tail current source; pseudo-differential circuits are not allowed	✓
Current Mirror Ratio	≤ 20	✓
Reference Current	Single ideal current source of arbitrary value, with positive node tied to VDD or negative node tied to GND	✓
CMFB circuit	Use ideal model provided	✓
Input voltage range	Measure	1.05 V
Pole/Zero cancellations and left half plane zeros	Forbidden	✓

Design Methodology:

Having to design an operational transconductance amplifier that must comply with a range of specifications imposed by the instructor can be quite daunting, especially for a student with little to no experience in circuit design. However, an effective way to alleviate this confusion is to scrupulously follow a practical design methodology. The gm/I_D design methodology was routinely studied in class to prepare students for this project, and I found this technique to be the most robust plan of attack to begin undertaking this design. After settling on a design methodology, it is important to select an appropriate amplifier topology.

Before choosing a topology, it is critical to refer to the given specifications in order to pick the most optimal solution. Since the amplifier needed to provide a very large open-loop gain of 76.5-dB while also offering a large output swing, I decided to implement a two-stage amplifier containing a telescopic cascode for the first stage, and a common source amplifier for the second stage. Ensuring the correct order of these two stages is paramount, as the telescopic cascode offers high gain with low output swing and positioning it after the common source amplifier would completely jeopardize the signal. Although I selected this topology, various other options were available to students.

Indeed, an alternative to the telescopic cascode would be to apply a folded cascode because of its ability to grant larger output swing. However, the folded configuration yields less gain than the telescopic setup, and it also adds an extra rail which increases the total power dissipation. Since it was primary for me to minimize the total power consumption while meeting the open-loop gain specification, I decided not to opt for the folded arrangement. Another option would be to use a cascade layout, with both stages being common source amplifiers.

Yet, a common source amplifier only provides a gain of $gmr_o/2$, and the total gain of the cascade format would provide a gain of $gmr_o/2 * gmr_o/2 = (gmr_o)^2/4$. It is therefore more desirable to use a telescopic cascode followed by a common source amplifier, since the gain of the telescopic cascode is $(gmr_o)^2/2$, and the total gain of the amplifier would be $(gmr_o)^2/2 * gmr_o/2 = (gmr_o)^3/4$, which is much larger than the total gain of the cascade structure.

Interconnection between Design Specifications and Circuit/Component Parameters:

The first step to take when writing up a design script is to mark down the assigned specifications, as done in lines 17-53 of the MATLAB code in Appendix A. Once the designing field is clearly laid out with specifications, one can begin to make educated assumptions to initiate the design process. For example, at line 59, I have assumed a gm/I_D value of 15 S/A. This is an ideal value to assume, as it suggests that the MOS devices are neither in the weak inversion region nor in strong inversion region, thus leaving room, for correction as the design progresses. Additionally, in order to facilitate the noise calculations, I assumed a gamma of 1, as shown in line 63. After having made vague assumptions, one can begin to work with general equations.

Indeed, the specified closed-loop gain of 2 combined with the closed-loop gain equation $A_V = C_S/C_F$ caused $C_S = 2C_F$, which in turn yielded a $\beta_{max} = 1/3$ as shown in line 60. Line 62 shows the value of β is obtained by multiplying β_{max} by a coefficient. After the closed-loop gain was related, the loop gain was addressed in line 76, where $T = 1/\epsilon_s$. Next, given the dynamic range specification, the noise and output swing were computed in lines 79-83. Notice that I added an extra gm/I_D at line 80. This was done simply by measure of precaution and to have some error margin for the overdrive voltage.

I then proceeded to design the first stage of the OTA, and the process is displayed in lines 86-164. Given the loop gain, I obtained the open-loop gain by dividing T by β at line 87. I then partitioned the gains between the telescopic cascode at the first stage and the common source amplifier at the second stage. Knowing that the telescopic cascode provides a much higher gain than the common source amplifier, I let the gain of the common source amplifier at the second stage only equal 10. I then related terms in lines 87-90 and got a stage 1 gain of 612.25 V/V.

Lines 92-95 show important parameters, especially $x_2 = C_{LTOT}/C_C$, which along with β, are main tuning knobs for this design. Afterwards, in lines 97-102, the compensation capacitor C_C was computed given the noise values, and C_{LTOT} was obtained by relating C_C to x_2. As shown in line 101, it is then trivial to obtain C_L from C_{LTOT}, x_3, and β. Following that, in lines 103-107, I included a conditional statement that verifies if $C_L < C_{LMIN}$, which is the minimum load capacitance given in the specifications. If that condition is true, then $C_L = C_{LMIN}$ and C_{LTOT} and C_C are recomputed accordingly. Once C_L is known, C_S, C_F, and C_{gg1} can be calculated as depicted in lines 109-111.

In order to calculate the transconductance GM_1 of the first stage, it is imperative to know the unity gain frequency ω_C. Line 117 shows the computation of the unity gain frequency thanks to the settling time t_s and to the dynamic settling error ϵ_d. ω_C can then be combined with C_C and β in line 120 to yield GM_1. The following line shows the transit frequency of the first stage, ω_{T1}.

The widths and lengths of the devices were then extracted in lines 125-161. First, I declared the lengths of the input and load MOS of the 1st stage, L_1 and L_{11} respectively. Then, using a for loop, I iterated through each input length and used the "whatis" function to look up the gm/I_D value, and then used that value along with GM_1 in line 133 to obtain the current I_{D1} flowing through that branch of the half-circuit. Within that first for loop iterating through values for the input length L_1, I nested another for loop in line 135 that iterates through values for L_{11}. In the for loop, I computed the open-loop gain of the first stage according to the following equation:

$$A_{o1} = \frac{\left(\frac{gm}{I_D}\right)_1^2}{\left(\frac{gds}{I_D}\right)_1^2 + \left(\frac{gds}{I_D}\right)_{11}^2}$$

This expression is a lot more convenient to use in MATLAB than the generic telescopic cascode gain equation described in the first section of this report because I can now simply use the "whatis" function to look up very accurate simulation values of $\left(\frac{gds}{I_D}\right)_1$ and $\left(\frac{gds}{I_D}\right)_{11}$.

I then wrote a conditional statement in line 142 that checks if the calculated gain is greater than the product of the specified gain and a margin coefficient. If that is the case, I then use the "whatisVGS" function in line 144 to seek out the value V_{GS1} for the input MOS. Notice that it is critical to specify "nch" when using the function, since the input device is an NMOS. Using that value for V_{GS1}, I can then lookup the current density J_{D1} to then obtain the width W_1 of the input transistor. A similar operation was conducted in lines 147-149 to obtain the width W_{11} of the PMOS load, only this time, I specified "pch" when using the lookup functions.

Once I obtained the width iterations W_1 and W_{11}, I declared a boolean variable "validity" in line 152, which checks if those widths comply with the design constraints. If those lengths are valid, then those widths are recorded, and the iteration is terminated by breaking out of the loops, as showed in lines 154 and 159. Once the process has broken out of the loops, I_{D1} is computed using GM_1 and the gm/I_D obtained in the iteration.

The beauty about this iteration is that the code begins computing for widths linked with the lowest possible lengths, since the length for loop is the highest-level loop. That means that if the widths W_1 and W_{11} of the first iteration comply with design specifications, then the valid lengths L_1 and L_{11} are of the lowest possible value of 180nm.

Now that the parameters of stage 1 have been calculated, we can move on to compute the parameters for the common source amplifier of the second stage. Line 169 shows a k value of 3, correlating with the phase margin value of 72 degrees. Given k, I was able to calculate the value of the second pole ω_{p2}. In line 171, C_{gg2} was obtained given the ratio $x_1 = C_{gg2}/C_C$. In lines 173-177, C_1 and C_2 were computed using C_{gg2} and C_{LTOT} along with the ratios θ_1 and θ_2. I then used the values of C_1 and C_2 in line 177 to obtain the transconductance of the second stage, GM_2. Finally, given GM_2 and C_{gg2}, I was able to extract the value of the transit frequency of the second stage ω_{T2}, as shown in line 179.

The process of obtaining the dimensions of the devices in the second stage is identical to that of the first stage, at the exception that the gain of the common source amplifier of the second stage is defined with the following expression:

$$A_{o2} = \frac{\left(\frac{gm}{I_D}\right)_2}{\left(\frac{gds}{I_D}\right)_2 + \left(\frac{gds}{I_D}\right)_{22}}$$

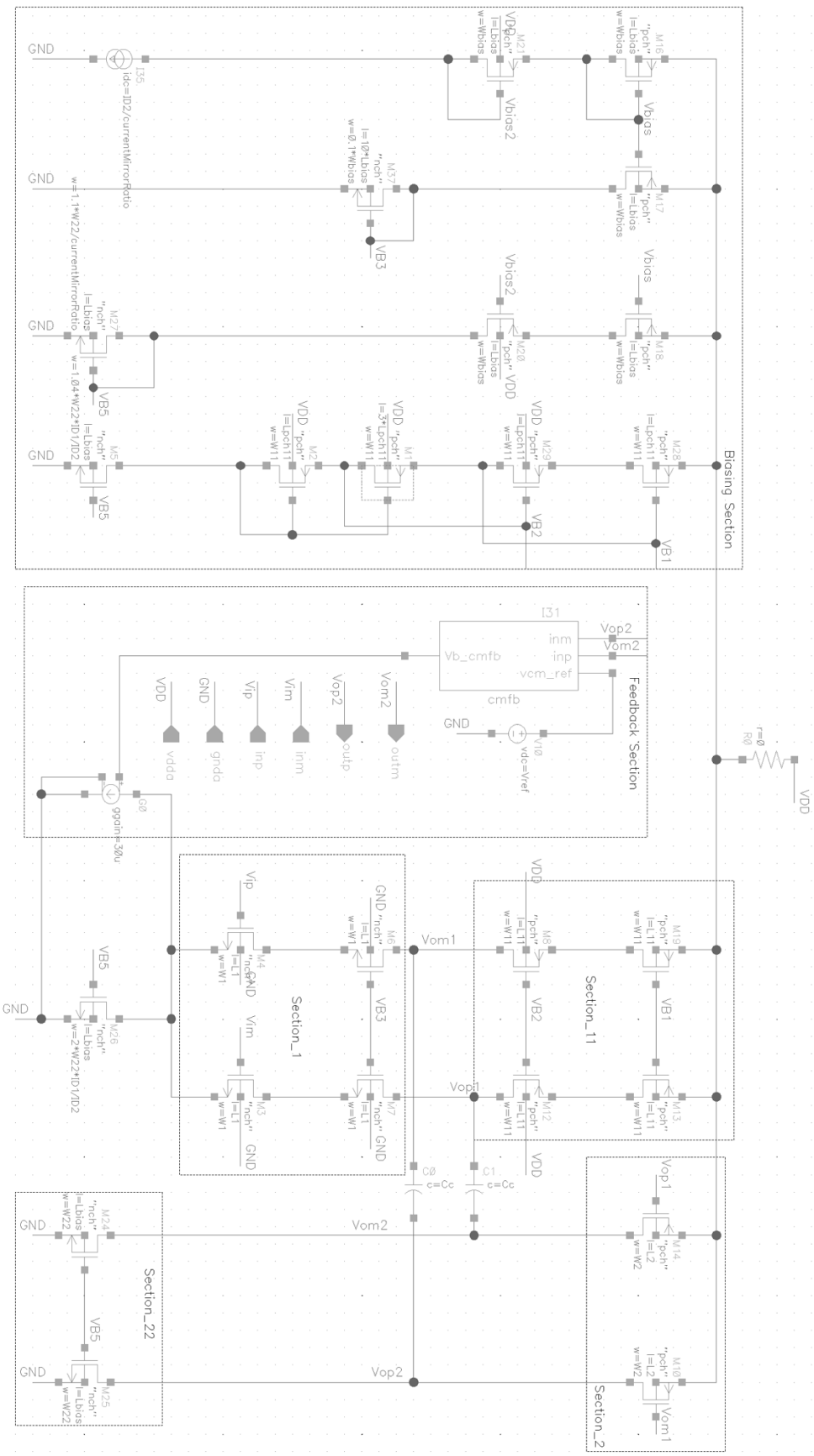
That being known, the nested for loops in lines 185-217 follow the exact same logic as the ones in lines 128-161.

The final step of this MATLAB script is to compute the minimum possible bias current given the maximum current mirror ratio that was constrained. I did that in line 220, where I took the maximum current branch, which was either the tail current of stage 1 (equal to 2 times I_{D1}) or one of the branches of the differential common source amplifier of stage 2, and divided it by the maximum current mirror ratio of 20.

Lines 229-269 are just used to print out results of importance for the overall ease of use of the script.

Figure A3 in appendix A shows another MATLAB script that was used to iterate the main script through values of x_2 and β, the two principal design knobs. Line 21 shows "parfor", which is a for loop that enables parallel computations to greatly reduce the overall computing time. In line 22, "beta_x2_function" refers to a function that uses the same code as the main design script described above. To finish, lines 25-27 are used to output a 3-dimensional surface plot of the computations. That surface plot is displayed below Figure A2, with the optimal value of x_2 and β that provide the lowest total current consumption in the two stages of the OTA. Those values of x_2 and β were used in the main script to extract the parameters of the transistors.

The parameter values obtained in MATLAB compare well with those obtained in Cadence. The MATLAB predicted total current consumption for the two stages (excluding the current needed for the biasing circuit) was 340.0 μA, and that simulated in Cadence was 336.4 μA, which is only 1.06% off from the calculated value. I am extremely pleased with the accuracy of this result. Additionally, I accounted for a phase margin of 72 degrees, and the simulated phase margin was 67.79 degrees, thus yielding a percentage difference of 5.85%. That result is acceptable as well. The rest of the important calculated parameters printed out in Figure A2 can be compared with the 2nd circuit schematic, but overall, all are in agreement.


Biasing Section
Feedback Section
Section_1
Section_11
Section_2
Section_22
GND
VDD
Vbias
Vbias2
VB1
VB2
VB3
VB5
Vop1
Vop2
Vom1
Vom2
Vip
Vim
inp
inm
outp
outm
vdda
gnda
cmfb
Vb_cmfb
vcm_ref
ggain=30u
vdc=Vref
idc=ID2/currentMirrorRatio
w=1.1*W22/currentMirrorRatio
w=1.04*W22*ID1/ID2
w=2*W22*ID1/ID2
r=0
R0
C0
C1
c=Cc

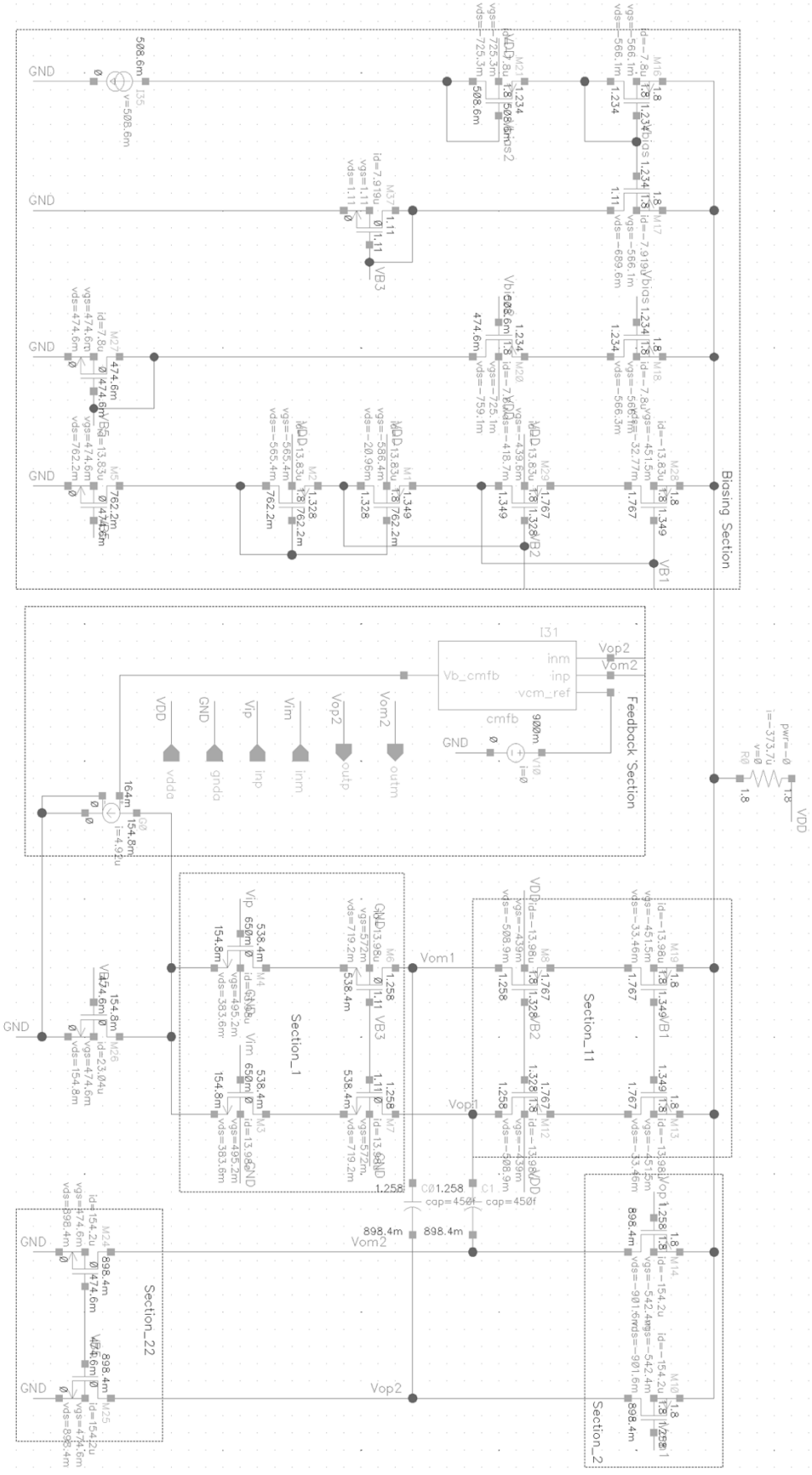
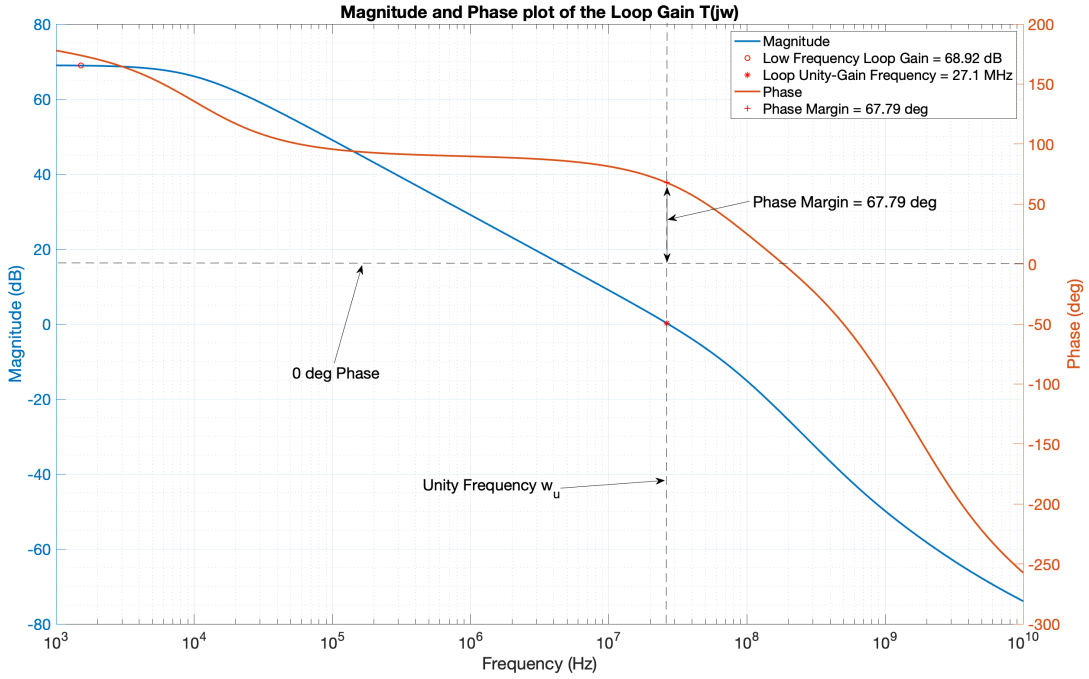
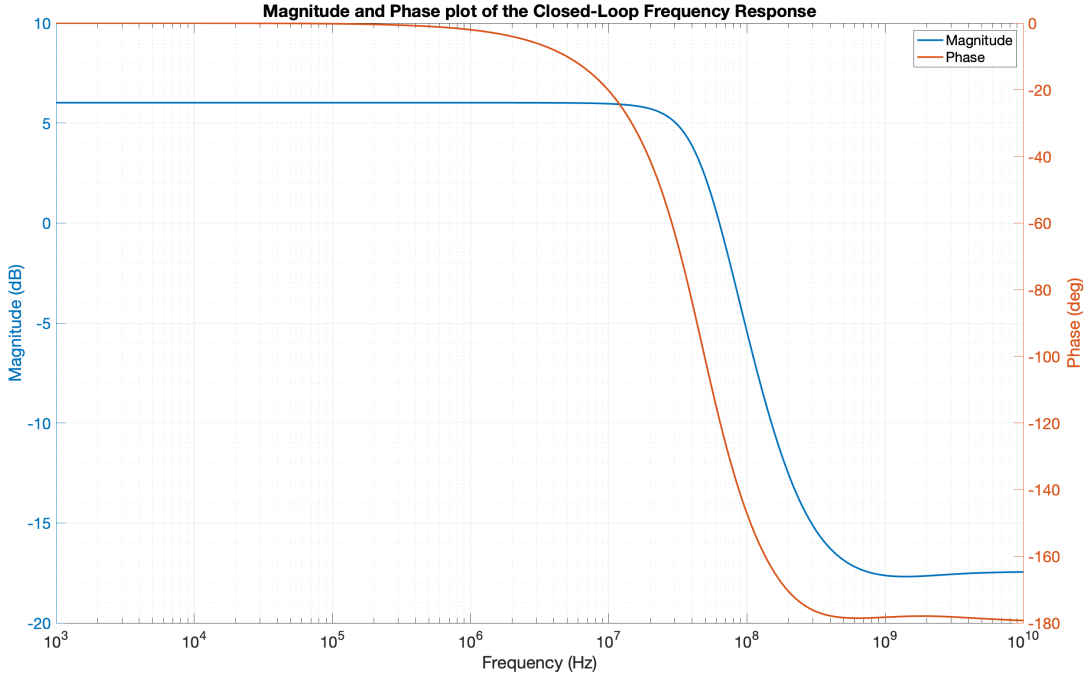
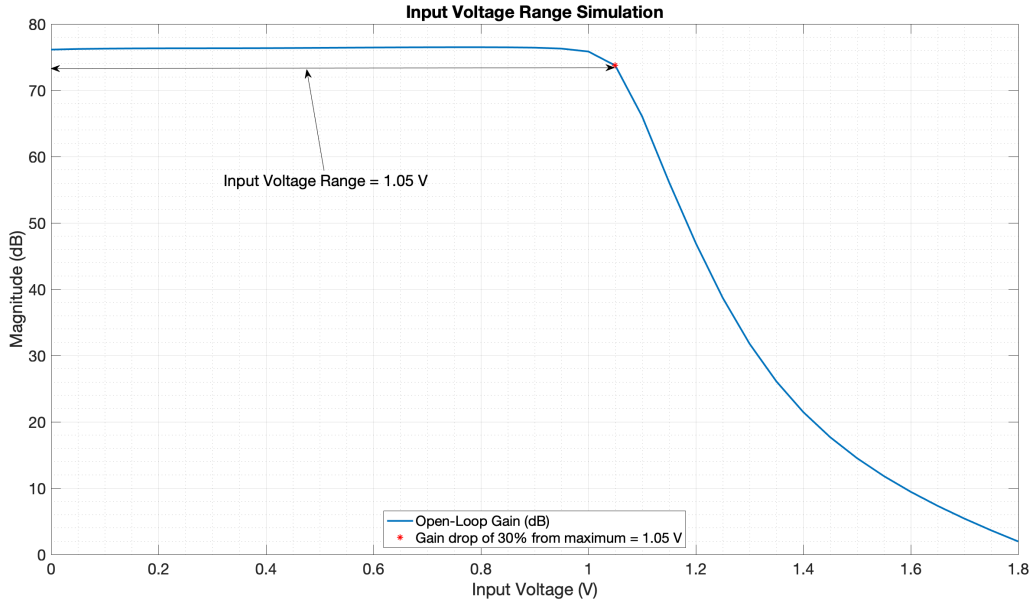

Biasing Section
Feedback Section
Section_11
Section_2
Section_1
Section_22
Vb_cmfb
inm
inp
vcm_ref
cmfb
Vop2
Vom2
Vop1
Vom1
VDD
GND
VB1
VB2
VB3
Vip
Vim
vdda
gnda
outp
outm
C0 1.258
C1 1.258
cap=450f
R0
pwr=0
i=-373.7u
v=0
I31
I35
I30
M0
M1
M2
M3
M4
M5
M6
M7
M10
M11
M12
M13
M14
M16
M17
M18
M19
M20
M21
M23
M24
M25
M26
M27
M28
M29
M37

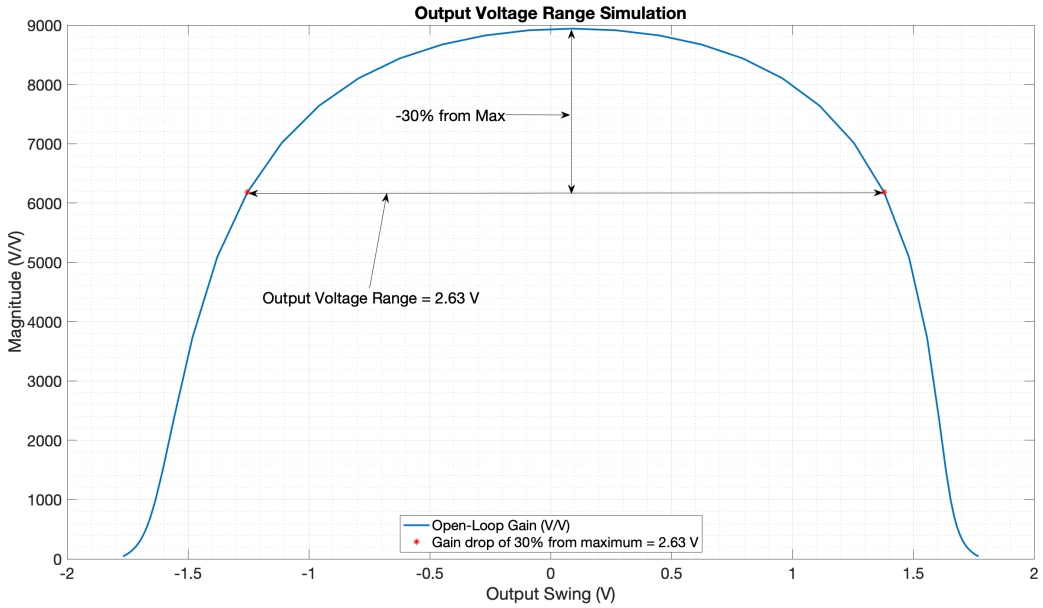
Table annotating the gm/I_D of the transistors in the schematic above

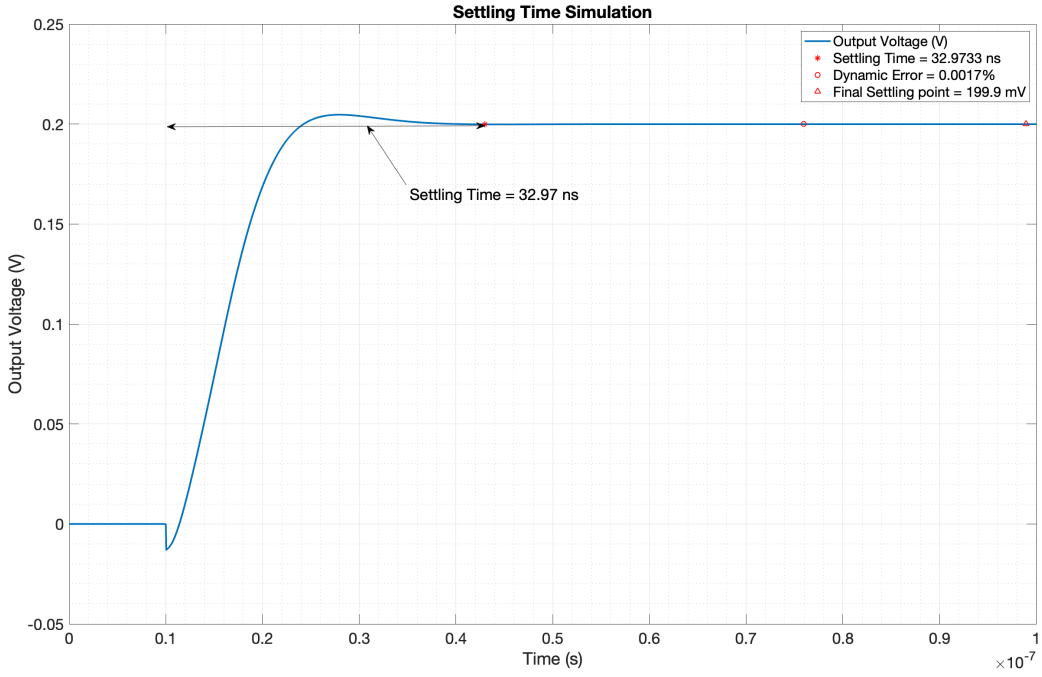
Transistor	gm/I_D (S/A)
M16	13.78
M21	13.58
M17	13.73
M37	2.67
M18	13.78
M20	13.59
M27	20.89
M28	22.51
M29	23.67
M1	18.69
M2	24.77
M5	20.69
M19	22.53
M8	23.67
M6	24.54
M4	23.88
M26	21.04
M13	22.53
M12	23.67
M7	24.54
M3	23.88
M14	20.56
M24	20.60
M10	20.56
M25	20.60

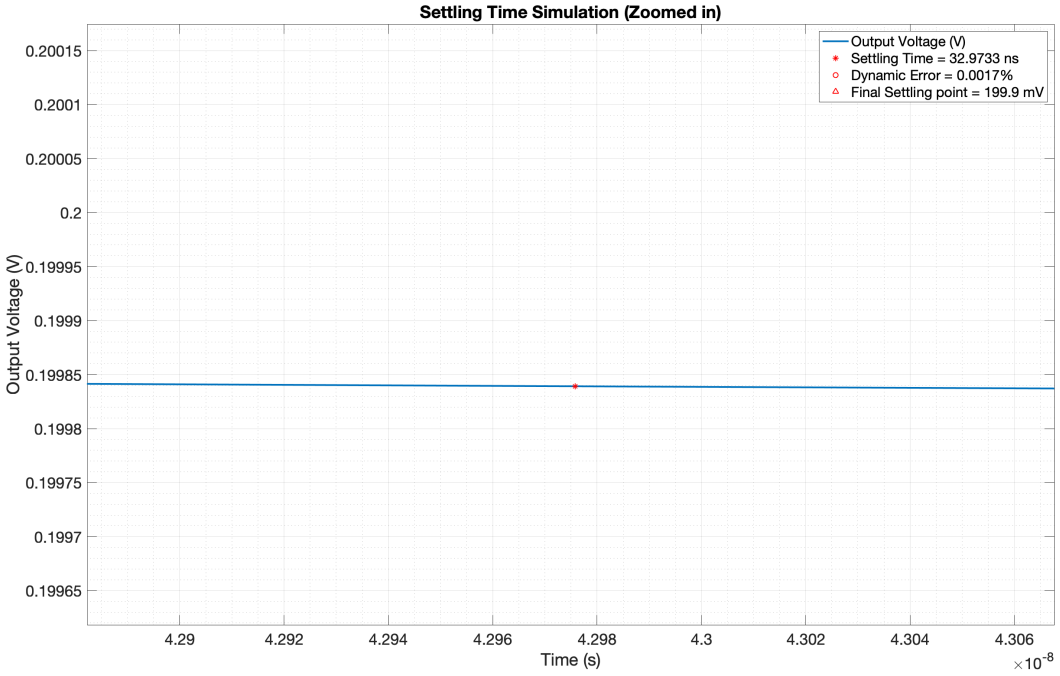

Magnitude and Phase plot of the Loop Gain T(jw)
Magnitude
Low Frequency Loop Gain = 68.92 dB
Loop Unity-Gain Frequency = 27.1 MHz
Phase
Phase Margin = 67.79 deg
Phase Margin = 67.79 deg
0 deg Phase
Unity Frequency w u
Magnitude (dB)
Phase (deg)
Frequency (Hz)

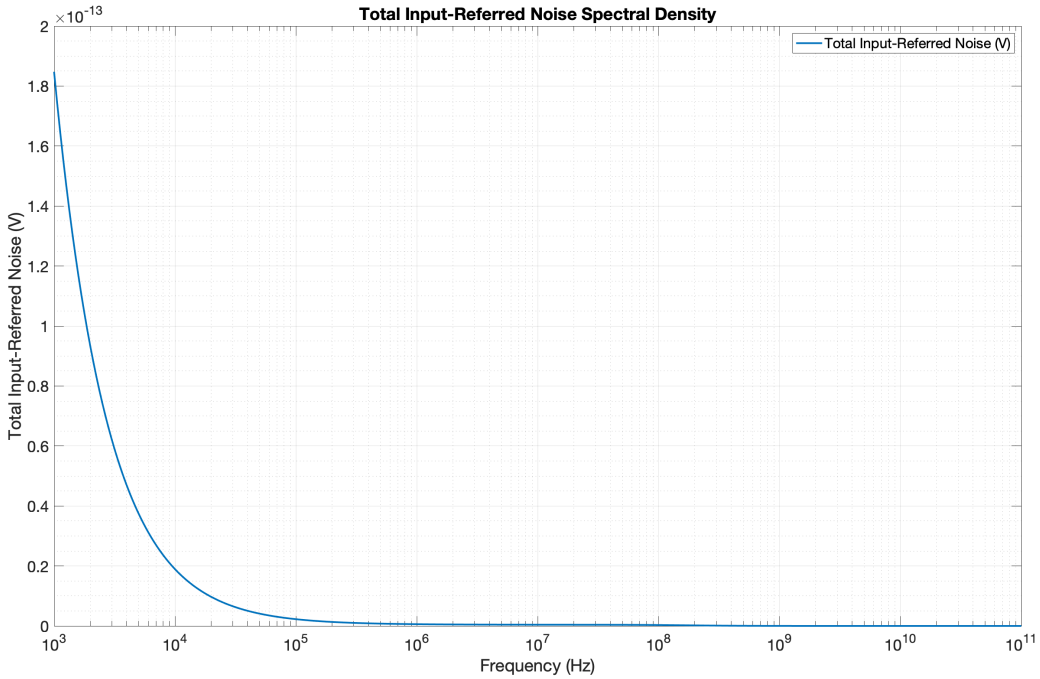

Magnitude and Phase plot of the Closed-Loop Frequency Response
Magnitude
Phase
Magnitude (dB)
Phase (deg)
Frequency (Hz)


Input Voltage Range Simulation
Input Voltage Range = 1.05 V
Magnitude (dB)
Input Voltage (V)
Open-Loop Gain (dB)
Gain drop of 30% from maximum = 1.05 V


Output Voltage Range Simulation
-30% from Max
Output Voltage Range = 2.63 V
Magnitude (V/V)
Output Swing (V)
Open-Loop Gain (V/V)
Gain drop of 30% from maximum = 2.63 V


Settling Time Simulation
Output Voltage (V)
Settling Time = 32.9733 ns
Dynamic Error = 0.0017%
Final Settling point = 199.9 mV
Settling Time = 32.97 ns
Output Voltage (V)
Time (s)
×10^-7


Settling Time Simulation (Zoomed in)
Output Voltage (V)
Settling Time = 32.9733 ns
Dynamic Error = 0.0017%
Final Settling point = 199.9 mV
Output Voltage (V)
Time (s)
×10^-8

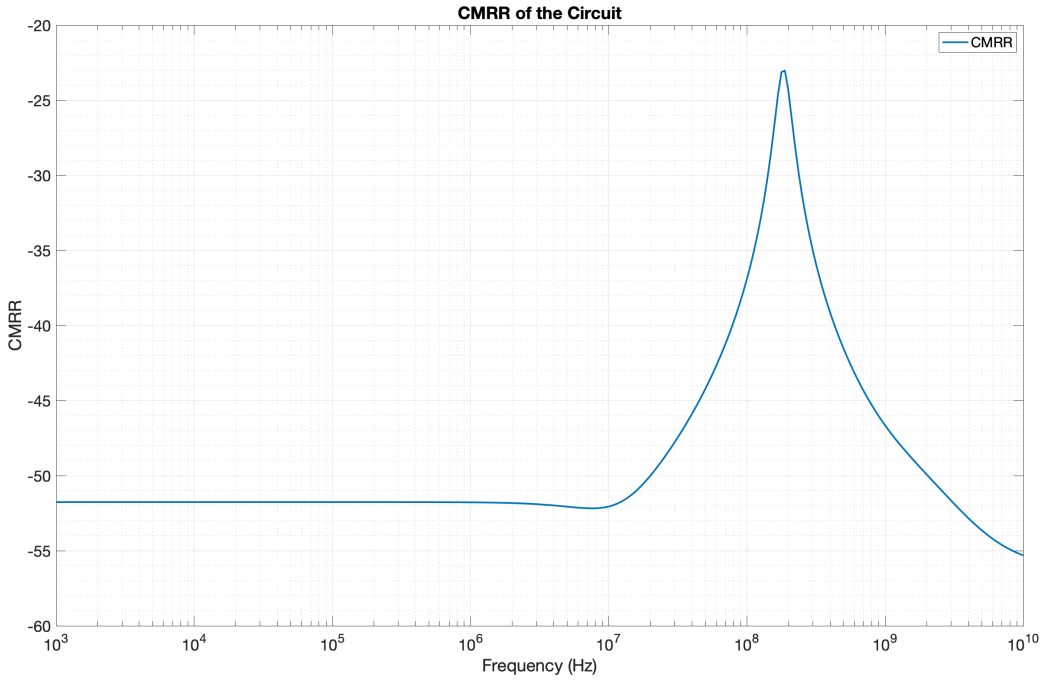


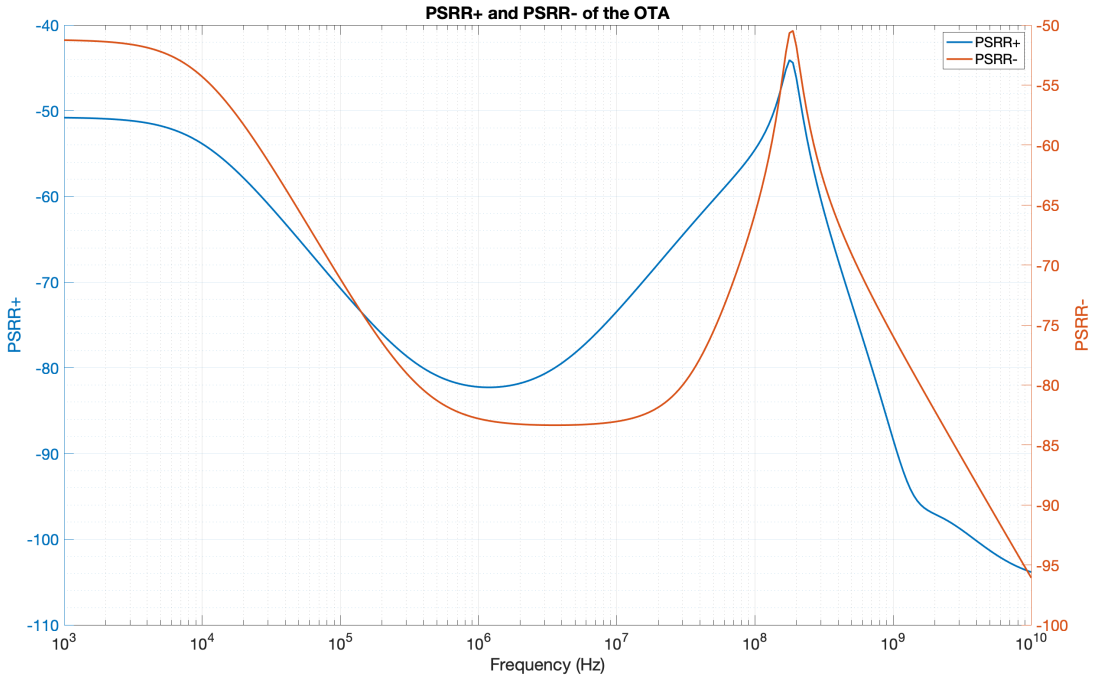
Cadence Report for the top 12 Noise Contributors

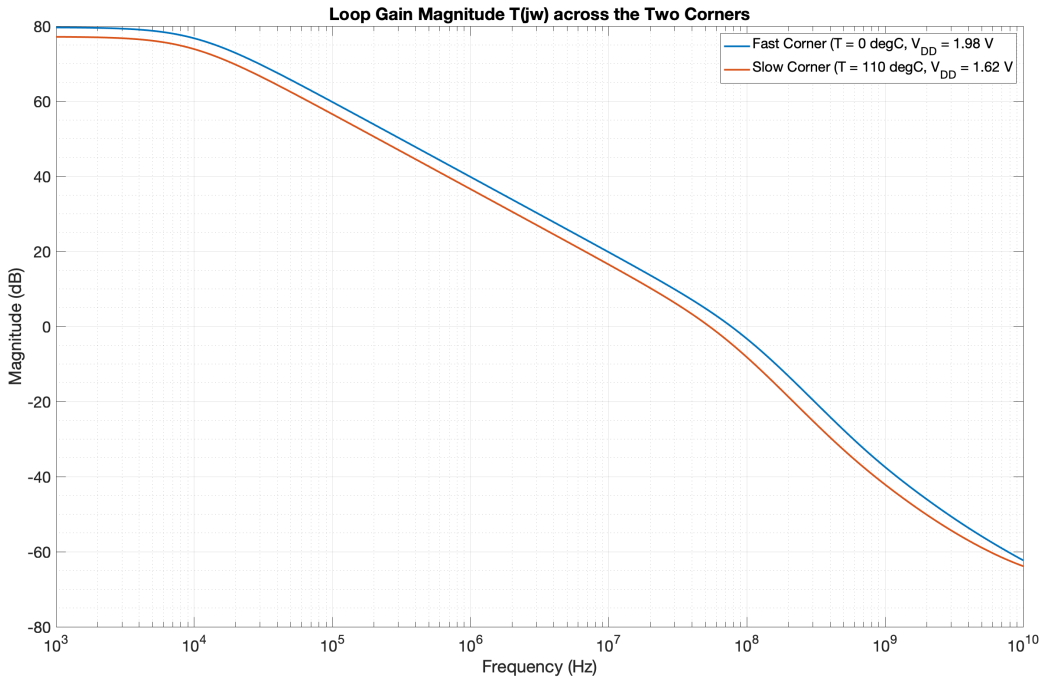
Window Expressions Info Help cadence

Device	Param	Noise Contribution	% Of Total
/I37/M13	id	1.85571e-08	20.66
/I37/M19	id	1.85571e-08	20.66
/I37/M4	id	1.75857e-08	19.58
/I37/M3	id	1.75857e-08	19.58
/I37/M4	fn	3.3908e-09	3.78
/I37/M3	fn	3.3908e-09	3.78
/I37/M12	id	2.32244e-09	2.59
/I37/M8	id	2.32244e-09	2.59
/I37/M24	id	1.38148e-09	1.54
/I37/M25	id	1.38148e-09	1.54
/I37/M14	id	1.07822e-09	1.20
/I37/M10	id	1.07822e-09	1.20

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Summarized Noise = 8.98007e-08
Total Input Referred Noise = 1.20541e-07
The above noise summary info is for noise data


CMRR of the Circuit
CMRR
-20
-25
-30
-35
-40
-45
-50
-55
-60
10^3
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}
Frequency (Hz)


PSRR+ and PSRR- of the OTA
PSRR+
PSRR-
-40
-50
-60
-70
-80
-90
-100
-110
-50
-55
-60
-65
-70
-75
-80
-85
-90
-95
-100
10^3
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}
Frequency (Hz)



Compliance Table specifying parameters at the Nominal Point and at both Corners

Parameter	Specification for EE338L	Specifications Achieved		
		Nominal	Fast Corner	Slow Corner
Technology	0.18µm process[1], N-well, nominal corner	✓	✓	✓
Supply Voltage	1.8 V (no scaling allowed)	1.8 V	1.98 V	1.62 V
Temperature	25°C	25°C	0°C	110°C
Power Dissipation	Minimize	672.66 µW	747.64 µW	594.38 µW
CL	≥ 2pF	✓	✓	✓
Closed-loop gain	2	1.9981	1.9994	1.9969

Cs	Cs ≤ CL	✓	✓	✓
Dynamic Range[2]	70 dB	75.78 dB	77.12 dB	72.58 dB
Static settling error	≤0.05%	0.0498%	0.0453%	0.0498%
Dynamic settling error	≤0.05%	0.0017%	0.0014%	0.0017%
Setting time[3]	35ns	32.97 ns	28.47 ns	46.33 ns
CMRR[4] at DC	>100dB	129.85 dB	130.25 dB	128.52 dB
PSRR[4] +/- at DC (10mV	>80dB	PSRR+ = 132.40 dB PSRR- = 131.95 dB	PSRR+ = 136.38 dB PSRR- = 136.25 dB	PSRR+ = 126.94 dB PSRR- = 126.23 dB
Passives	Total ideal capacitance < 10pF; Total ideal resistance < 100KΩ	✓	✓	✓
OTA input stage	NMOS or PMOS differential pair with a tail current source; pseudo-differential circuits are not allowed	✓	✓	✓
Current Mirror Ratio	≤ 20	✓	✓	✓
Reference Current	Single ideal current source of arbitrary value,	✓	✓	✓
CMFB circuit	Use ideal model provided	✓	✓	✓
Input voltage range	Measure	1.05 V	1.03 V	1.19 V
Pole/Zero cancellations and left half plane zeros	Forbidden	✓	✓	✓

Appendix A:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script offers a rough attempt to design the OTA using the initial
% specs.
%
% Analog IC Design --- EE 382M-14
% Written by Thomas Plantin
% Monday November 12th, 2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;
clear;
load 180nmos.mat; load 180pmos.mat;




%-----------------------------INITIAL SPECS---------------------------------%

% Technology --> 0.18um
% Temperature --> 25degC
% Power Dissipation --> MINIMIZE

Vdd = 1.8;                %Supply voltage
Av = 2;                   %Closed-loop gain
CL_min = 2E-12;           %CL >=2pF;
%Cs <= CL;
DR = 70;                  %Dynamic Range (in dB)
epsilon_s = 0.0005;       %Static settling error < 0.05%
epsilon_d = 0.0005;       %Dynamic settling error < 0.05%
ts = 35E-9;               %Setting time
%CMRR > 100dB;
%PSSR_p > 80dB;
%PSSR_m > 80dB;

% Passives --> Total ideal capacitance < 10pF;
%              Total ideal resistance < 100kOhm

% OTA Input Stage --> NMOS or PMOS differential pair with a tail current
```

Figure A1: MATLAB design script

```
39      %                       current source
40
41      % Current Mirror Ratio <= 20
42
43      % Reference Current --> Single ideal current source with arbitrary value,
44      %                        with positive node tied to Vdd or negative node
45      %                        tied to GND.
46
47      % CMFB Circuit --> Use ideal model provided.
48
49      % Input Voltage Range --> Measure.
50
51      % Pole/Zero cancellations and left half plane zeros --> Forbidden
52
53      %-------------------------------------------------------------------------%
54
55
56
57      %-------------------------TENTATIVE VARIABLES-----------------------------%
58
59 -    gm_ID = 15;
60 -    beta_max = 1/3;
61 -    beta_coeff = 0.98;
62 -    beta = beta_coeff*beta_max;
63 -    gamma = 1;
64 -    Temp = 25 + 273.15;    %25C to Kelvin
65 -    kb = 1.38065E-23;   %Boltzmann's Constant
66 -    PM = 72;            %Phase Margin = 72 deg = tan^-1(k), yielding k = 3
67
68
69      %-------------------------------------------------------------------------%
70
71
72
73      %----------------------------CALCULATION----------------------------------%
74
75      % Calculate loop gain from static settling error
76 -    T = 1/epsilon_s;
```

Figure A1 (continued): MATLAB design script

```
77
78
79      % Dynamic Range (DR), noise and swing
80 -    Vov = 2/gm_ID + 2/gm_ID; %Added an overdrive for margin
81 -    vodpeak = Vdd - 2*Vov;                %Assume a pk output voltage of 1.2V
82 -    vodntot = vodpeak^2 / 10^(DR/10);    %vodntot^2
83 -    vodntot_RMS = sqrt(vodntot);
84
85
86      %-FIRST STAGE-----------------------------------------------------------------
87 -    Ao = T./beta;
88 -    Ao2 = 10;              %Second stage - gain of 10
89 -    Ao1 = Ao/Ao2;          %Ao = Ao1*Ao2 First stage - gain of 800
90 -    over_design_coeff = 1.2;
91
92      % Gain of the first stage is equal to ~ (gmro)^2/2
93 -    x1 = 0.6;                                  %x1 = Cgg2/Cc
94 -    x2 = 4; % linspace(0.5, 1.5);              %x2 = CLtot/Cc
95 -    x3 = 1.25;                                 %x3 = CL/Cs
96
97 -    G1 = 1;   %G1 = gm11/gm1 - 1st stage
98 -    G2 = 1;   %G2 = gm22/gm2 - 2nd stage
99 -    x=1;
100 -   Cc = 2.*(kb.*Temp./vodntot).*(gamma./beta.*(1+G1)+1./x2.*(1+gamma.*(1+G2)));
101 -   CLtot = Cc .* x2;
102 -   CL = CLtot*2*x3/(2*x3+1-beta);
103
104 -   if CL<CL_min
105 -       CL = CL_min;
106 -       CLtot = CL*(2*x3+1-beta)/(2*x3);
107 -       Cc = CLtot / x2;
108 -   end
109
110 -   Cs = CL/x3;
111 -   Cf = Cs/2;
112 -   Cgg1 = Cf./beta - Cf - Cs;
113
114 -   slewCoeff = 1.3;
```

Figure A1 (continued): MATLAB design script

```
ts = slewCoeff * ts;

% Calculate fc given setting time and dynamic settling error
wc = (-1/ts)*log(epsilon_d);
fc = wc/2/pi;

GM1 = wc*Cc./beta;
wt1 = GM1./Cgg1;
wt1 = max(wt1, 10*wc);
ft1 = wt1/(2*pi);

L_1 = 0.18:0.02:2;          %NMOS Length of stage 1
L_11 = 0.18:0.02:2;         %PMOS Length of stage 1

for i = 1:length(L_1)

    L1 = L_1(i);
    %find the gm/id for the design
    gm_ID_real_1 = whatis(nch, 'GM_ID', 'GM_CGG', wt1,'L',L1);
    ID1 = GM1./gm_ID_real_1;

    for j = 1:i     %From 1 to i to ensure that PMOS device is smaller than NMOS device

        L11 = L_11(j);
        GDS_ID_1 =  whatis(nch, 'GDS_ID', 'GM_ID', gm_ID_real_1, 'L', L1);
        GDS_ID_2 =  whatis(pch, 'GDS_ID', 'GM_ID', gm_ID_real_1, 'L', L11);
        A1_calc = (gm_ID_real_1.^2)./(GDS_ID_1.^2+GDS_ID_2.^2);

        if A1_calc > over_design_coeff * Ao1

            VGS_1 = whatisVGS(nch, 'GM_ID', gm_ID_real_1, 'VDS', (Vdd - (2./gm_ID_real_1))./4, 'L', L1);
            JD_1 = whatis(nch, 'ID', 'VGS', VGS_1, 'VDS', (Vdd - (2./gm_ID_real_1))./4, 'L', L1)/nch.W;
            W1 = ID1/JD_1;
            VGS_11 = whatisVGS(pch, 'GM_ID', gm_ID_real_1, 'VDS', (Vdd - (2./gm_ID_real_1))./4, 'L', L11);
            JD_11 = whatis(pch, 'ID', 'VGS', VGS_11, 'VDS',(Vdd - (2./gm_ID_real_1))./4, 'L', L11)/pch.W;
            W11 = ID1./JD_11;
```

Figure A1 (continued): MATLAB design script

```
            validity = (W1>0.24 && W1<500) && (W11>0.24 && W11<500);
            if validity
                break;
            end
        end
    end
    if validity
        break;
    end
end

ID1 = GM1/gm_ID_real_1;


%%%%%%%%% SECOND STAGE CALCULATIONS %%%%%%%%%%%%%%

k = 3;               %pole coefficent for required phase margin
wp2 = k*wc;     %second pole value
Cgg2 = x1*Cc;

theta1 = 0.3;
theta2 = 0.3;
C1 = Cgg2*(1+theta1);
C2 = CLtot*(1+theta2);
GM2 = wp2*(C1*C2/Cc+C1+C2);

wt2 = GM2/Cgg2;
wt2 = max(wt2,10*wc);
ft2 = (1/2/pi)*wt2;
%FIND LENGTHS
L_2 = 0.18:0.02:2;          %NMOS Length of stage 2
L_22 = 0.6;         %PMOS Length of stage 2
for i = 1:length(L_2)

    L2 = L_2(i);
    %find the gm/id for the design
    gm_ID_real_2 = whatis(pch, 'GM_ID', 'GM_CGG', wt2,'L',L2);
```

Figure A1 (continued): MATLAB design script

```
    ID2 = GM2/gm_ID_real_2;

    for j = i:length(L_22)     %Start from i to ensure that PMOS device is smaller than NMOS device

        L22 = L_22(j);
        GDS_ID_1 =  whatis(pch, 'GDS_ID', 'GM_ID', gm_ID_real_2, 'L', L2);
        GDS_ID_2 =  whatis(nch, 'GDS_ID', 'GM_ID', gm_ID_real_2, 'L', L22);
        A2_calc = (gm_ID_real_2)/(GDS_ID_1+GDS_ID_2);

        if A2_calc > over_design_coeff * Ao2

            VGS_2 = whatisVGS(pch, 'GM_ID', gm_ID_real_2, 'VDS', Vdd/2, 'L', L2);
            JD_2 = whatis(pch, 'ID', 'VGS', VGS_2, 'VDS', Vdd/2, 'L', L2)/pch.W;
            W2 = ID2/JD_2;
            VGS_22 = whatisVGS(nch, 'GM_ID', gm_ID_real_2, 'VDS', Vdd/2, 'L', L22);
            JD_22 = whatis(nch, 'ID', 'VGS', VGS_22, 'VDS',Vdd/2, 'L', L22)/nch.W;
            W22 = ID2/JD_22;

            validity = (W2>0.24 && W2<500) && (W22>0.24 && W22<500);
            if validity
                break;
            end
        end
    end
    if validity
        break;
    end
end

IDtot = ID1 + ID2;
Ibias = max(2*ID1, ID2)/20;
ID1tot = 2*ID1;


W0 = 8*2*ID1/Ibias;


%--------------------------------------------------------------------------%
```

Figure A1 (continued): MATLAB design script

```
229
230     %------------------------------PRINTINGS-----------------------------------%
231
232 -   fprintf('Open-loop gain --> Ao = %d\n', Ao);
233 -   fprintf('Loop gain --> T = %d\n', T);
234 -   fprintf('Closed-loop gain --> Av = %d\n', Av);
235 -   fprintf('\n');
236
237 -   fprintf('Crossover Frequency (unity gain f) --> wc = %d rad/s\n', wc);
238 -   fprintf('Crossover Frequency (unity gain f) --> fc = %d Hz\n', fc);
239 -   fprintf('\n');
240
241 -   fprintf('Capacitors --> Cc = %d F\n', Cc);
242 -   fprintf('           --> CLtot = %d F\n', CLtot);
243 -   fprintf('           --> Cf = %d F\n', Cf);
244 -   fprintf('           --> Cs = %d F\n', Cs);
245 -   fprintf('           --> Cgg1 = %d F\n', Cgg1);
246 -   fprintf('           --> Cgg2 = %d F\n', Cgg2);
247 -   fprintf('\n');
248
249 -   fprintf('1st Stage: Transconductance --> GM1 = %d S\n', GM1);
250 -   fprintf('           Efficiency --> GM_ID1 = %d V^-1\n', gm_ID_real_1);
251 -   fprintf('           Speed --> wt1 = %d rad/s\n', wt1);
252 -   fprintf('           Current --> ID1 = %d\n', ID1);
253 -   fprintf('\n');
254
255 -   fprintf('2nd Stage: Transconductance --> GM2 = %d S\n', GM2);
256 -   fprintf('           Efficiency --> GM_ID2 = %d V^-1\n', gm_ID_real_2);
257 -   fprintf('           Speed --> wt2 = %d rad/s\n', wt2);
258 -   fprintf('           Current --> ID2 = %d\n', ID2);
259 -   fprintf('\n');
260
261 -   fprintf('Total current --> IDtot = %d\n', IDtot);
262 -   fprintf('\n');
263
264 -   fprintf('Bias current --> Ibias = %d\n', Ibias);
265 -   fprintf('\n');
266
267 -   fprintf('Stage 1 current --> ID1tot = %d\n', ID1tot);
268 -   fprintf('Stage 2 current --> ID2 = %d\n', ID2);
269 -   fprintf('\n');
270     %--------------------------------------------------------------------------%
```

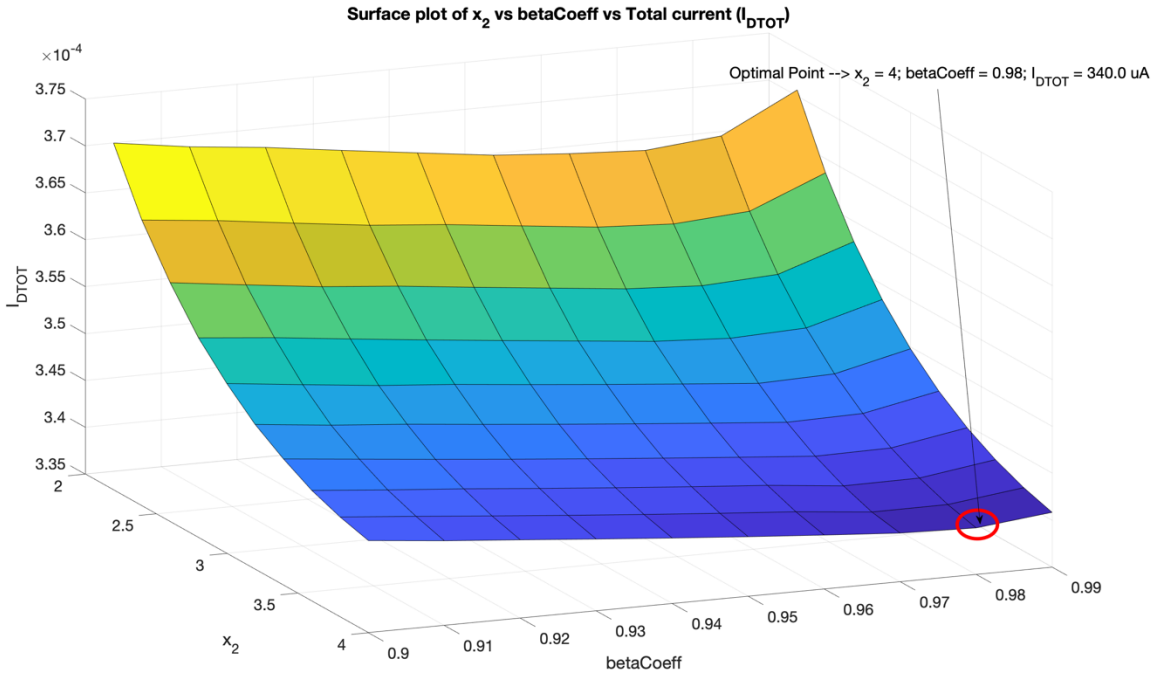
Figure A1 (continued): MATLAB design script

```
Command Window
  Open-loop gain --> Ao = 6.122449e+03
  Loop gain --> T = 2000
  Closed-loop gain --> Av = 2

  Crossover Frequency (unity gain f) --> wc = 1.670528e+08 rad/s
  Crossover Frequency (unity gain f) --> fc = 2.658728e+07 Hz

  Capacitors --> Cc = 6.346667e-13 F
             --> CLtot = 2.538667e-12 F
             --> Cf = 8.000000e-13 F
             --> Cs = 1.600000e-12 F
             --> Cgg1 = 4.897959e-14 F
             --> Cgg2 = 3.808000e-13 F

  1st Stage: Transconductance --> GM1 = 3.245597e-04 S
             Efficiency --> GM_ID1 = 2.340756e+01 V^-1
             Speed --> wt1 = 6.626428e+09 rad/s
             Current --> ID1 = 1.386559e-05

  2nd Stage: Transconductance --> GM2 = 3.192136e-03 S
             Efficiency --> GM_ID2 = 2.044786e+01 V^-1
             Speed --> wt2 = 8.382710e+09 rad/s
             Current --> ID2 = 1.561110e-04

  Total current --> IDtot = 3.399532e-04

  Bias current --> Ibias = 7.805551e-06

  Stage 1 current --> ID1tot = 2.773118e-05
  Stage 2 current --> ID2 = 1.561110e-04

fx >>
```

Figure A2: Output Print from the MATLAB script above

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script iterates through possible combinations of x2 and beta to find
% the optimal operating point that consumes the least total current.
%
% Analog IC Design --- EE 382M-14
% Written by Thomas Plantin
% Tuesday November 27th, 2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
clc;
load 180nmos.mat; load 180pmos.mat;


pch.ID = abs(pch.ID);

beta_sweep = 0.9:0.01:0.99;
x2_sweep = 2.2:0.2:4;

for i = 1:length(beta_sweep)
    parfor j = 1:length(x2_sweep)     %parallel for loop
        [IDtot(i,j), validity(i,j), lengths{i,j}, widths{i,j}] = beta_x2_function(nch, pch, beta_sweep(i), x2_sweep(j));
    end
end
surf(x2_sweep, beta_sweep, IDtot);
title('Surface plot of x2 vs beta_coeff vs Total current (I_D)');
xlabel('x2'); ylabel('beta'); zlabel('IDtot');
```

Figure A3: MATLAB iteration script to optimize the operating point



Appendix B:

```
// Generated for: spectre
// Generated on: Dec  5 17:25:05 2018
// Design library name: EE382M_EE338L_Project_Testbench
// Design cell name: Universal_Testbench_Top
// Design view name: schematic
simulator lang=spectre
global 0 vdd!
parameters Bias_vcm_in=650m CL=2p Cc=450f Cf=800f Cs=1.6p DC_Sweep=0 \
    ID1=13.8u ID2=156u Input_Offset=1.125u L1=180n L11=180n L2=180n \
    Lbias=600n Lpch11=180n Vref=900m Vstep=100m W1=22.8u W11=239.4u \
    W2=185u W22=194.8u Wbias=8u Wfeed=28u currentMirrorRatio=20 sim_CMRR=0 \
    sim_PSRR_m=0 sim_PSRR_p=0 sim_ac=1 sim_dcsweep=0 sim_noise=0 sim_stb=0 \
    sim_tran=0
include "/home/ecelrc/students/tplantin/cadence/ee382m_tsmc180n.scs"

// Library name: EE382M_EE338L_Project_Testbench
// Cell name: cmfb
// View name: schematic
subckt cmfb Vb_cmfb inm inp vcm_ref
    E2 (Vb_cmfb 0 net5 vcm_ref) vcvs gain=-100
    E1 (net014 0 inm 0) vcvs gain=0.5
    E0 (net5 net014 inp 0) vcvs gain=0.5
ends cmfb
// End of subcircuit definition.

// Library name: EE382M_EE338L_Project_Testbench
// Cell name: OTA_Design_2018
// View name: schematic
subckt OTA_Design_2018 GND Vim Vip Vom2 Vop2 VDD
```

I35 (Vbias2 GND) isource dc=ID2/currentMirrorRatio type=dc

G0 (net13 GND net23 GND) vccs gm=30u

M5 (net05 VB5 GND GND) nch w=1.04*W22*ID1/ID2 l=Lbias

M7 (Vop1 VB3 net31 GND) nch w=W1 l=L1

M6 (Vom1 VB3 net39 GND) nch w=W1 l=L1

M4 (net39 Vip net13 GND) nch w=W1 l=L1

M3 (net31 Vim net13 GND) nch w=W1 l=L1

M37 (VB3 VB3 GND GND) nch w=0.1*Wbias l=10*Lbias

M24 (Vom2 VB5 GND GND) nch w=W22 l=Lbias

M27 (VB5 VB5 GND GND) nch w=1.1*W22/currentMirrorRatio l=Lbias

M25 (Vop2 VB5 GND GND) nch w=W22 l=Lbias

M26 (net13 VB5 GND GND) nch w=2*W22*ID1/ID2 l=Lbias

I31 (net23 Vop2 Vom2 net24) cmfb

C0 (Vom1 Vop2) capacitor c=Cc

C1 (Vop1 Vom2) capacitor c=Cc

V10 (net24 GND) vsource dc=Vref type=dc

R0 (net08 VDD) resistor r=0

M28 (net25 VB1 net08 net08) pch w=W11 l=Lpch11

M14 (Vom2 Vop1 net08 net08) pch w=W2 l=L2

M13 (net27 VB1 net08 net08) pch w=W11 l=L11

M10 (Vop2 Vom1 net08 net08) pch w=W2 l=L2

M19 (net28 VB1 net08 net08) pch w=W11 l=L11

M16 (Vbias Vbias net08 net08) pch w=Wbias l=Lbias

M29 (VB1 VB2 net25 VDD) pch w=W11 l=Lpch11

M21 (Vbias2 Vbias2 Vbias VDD) pch w=Wbias l=Lbias

M20 (VB5 Vbias2 net034 VDD) pch w=Wbias l=Lbias

M18 (net034 Vbias net08 net08) pch w=Wbias l=Lbias

M12 (Vop1 VB2 net27 VDD) pch w=W11 l=L11

M8 (Vom1 VB2 net28 VDD) pch w=W11 l=L11

M17 (VB3 Vbias net08 net08) pch w=Wbias l=Lbias

M1 (VB2 net05 VB1 VDD) pch w=W11 l=3*Lpch11

```
   M2 (net05 net05 VB2 VDD) pch w=W11 l=Lpch11
ends OTA_Design_2018
// End of subcircuit definition.

// Library name: EE382M_EE338L_Project_Testbench
// Cell name: Universal_Testbench_Top
// View name: schematic
V5 (net73 0) vsource dc=DC_Sweep*sim_dcsweep+Input_Offset*sim_PSRR_p + Input_Offset*sim_PSRR_m \
        type=pwl mag=sim_ac wave=[ 10n 0 10.05n Vstep ]
R1 (inp p) resistor r=1G isnoisy=no
R0 (inm m) resistor r=1G isnoisy=no
W17 (net0176 outm noise_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T \
        rclosed=1m
W8 (net0178 outp noise_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T \
        rclosed=1m
W11 (inp p dcsweep_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W1 (inm m ac_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W0 (inp p ac_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W16 (inp p PSRR_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W15 (inm m PSRR_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W14 (net0176 outm tran_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T \
        rclosed=1m
W13 (inp p CMRR_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W6 (net0178 outp tran_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T \
        rclosed=1m
W12 (inm m CMRR_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W22 (net0178 outp stb_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T \
        rclosed=1m
W10 (inm m dcsweep_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T rclosed=1m
W23 (net0176 outm stb_sim 0) relay vt1=500.0m vt2=510.00m ropen=1T \
```

```
    rclosed=1m
I57 (out1p out1m outp outm) diffstbprobe
I37 (net0121 m p out1m out1p vdd!) OTA_Design_2018
V11 (net0121 0) vsource dc=0 mag=sim_PSRR_m type=dc
V6 (vdd! 0) vsource dc=1.8 mag=sim_PSRR_p type=dc
V10 (tran_sim 0) vsource dc=sim_tran type=dc
V4 (noise_sim 0) vsource dc=sim_noise type=dc
V9 (PSRR_sim 0) vsource dc=sim_PSRR_p + sim_PSRR_m type=dc
V2 (vcm_in 0) vsource dc=Bias_vcm_in mag=sim_CMRR type=dc
V3 (ac_sim 0) vsource dc=0 mag=sim_ac type=dc
V8 (CMRR_sim 0) vsource dc=sim_CMRR type=dc
V7 (dcsweep_sim 0) vsource dc=sim_dcsweep type=dc
V12 (stb_sim 0) vsource dc=sim_stb type=dc
E4 (inm vcm_in net73 0) vcvs gain=-0.5
E3 (inp vcm_in net73 0) vcvs gain=0.5
C3 (net0176 p) capacitor c=Cf
C2 (net0178 m) capacitor c=Cf
C6 (outp 0) capacitor c=3p
C5 (outm 0) capacitor c=3p
C9 (m inm) capacitor c=Cs
C10 (p inp) capacitor c=Cs
simulatorOptions options reltol=1e-3 vabstol=1e-6 iabstol=1e-12 temp=25.0 \
    tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 maxnotes=5 maxwarns=5 \
    digits=5 cols=80 pivrel=1e-3 sensfile="../psf/sens.output" \
    checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
```

primitives info what=primitives where=rawfile

subckts info what=subckts where=rawfile

saveOptions options save=allpub

include \

"/misc/linuxws/packages/cadence_2017/ic617/tools/dfII/etc/cdslib/artist/analogLib/diffstbprobe/diffstbprobe.scs"