



CMOS Analog IC Design

Project 01
Two-Stage Miller OTA

Supervised by: Dr. Hesham Omran

Part 1: gm/ID Design Charts

Using ADT Device Xplore, plot the following design charts vs gm/ID for both PMOS and NMOS. Set VDS = VDD/3 = 0.6 and L = 0.28u, 0.4u:0.4u:2u

- 1) gm/gds
- 2) ID/W
- 3) gm/Cgg (use advanced Y expression)
- 4) VGS

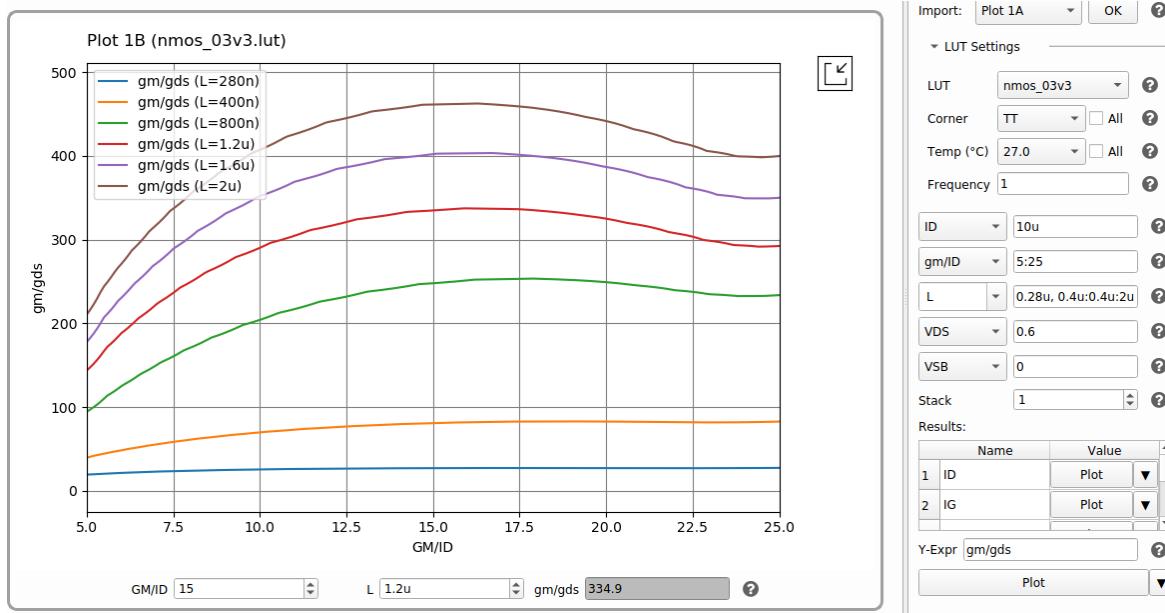


Figure 1 : gm/gds vs gm/id for Nmos

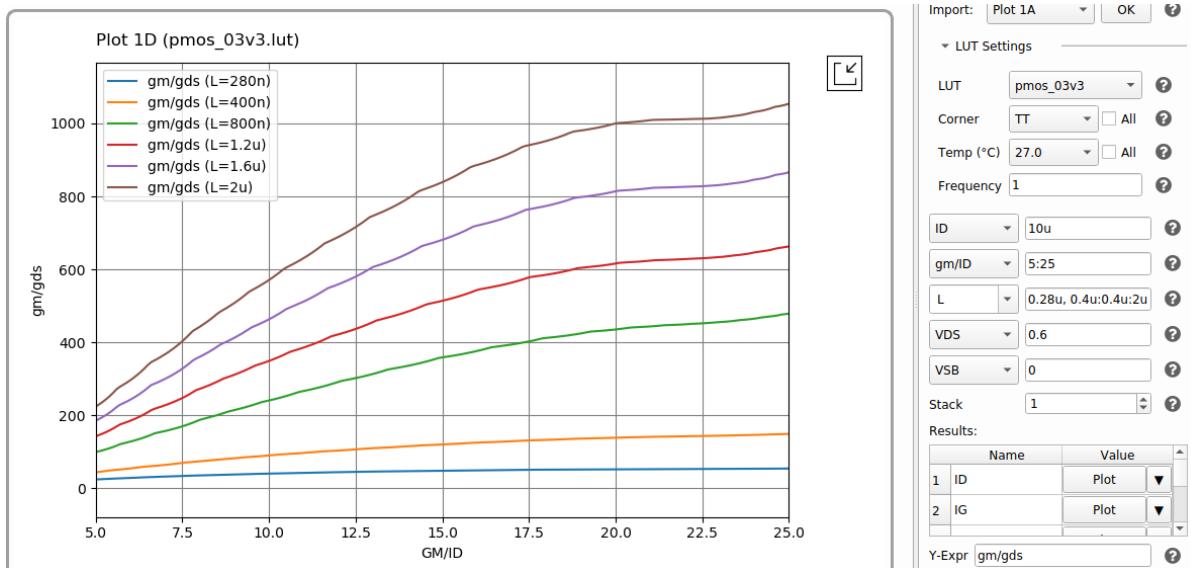


Figure 2 : gm/gds vs gm/id for Pmos

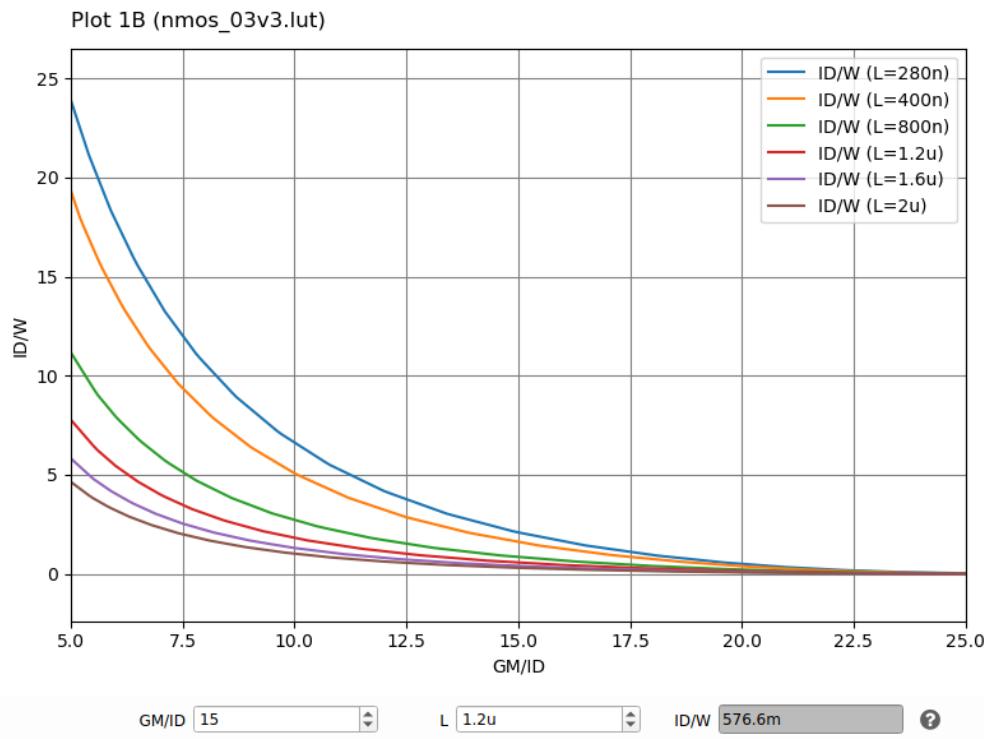


Figure 3 : : ID/W vs gm/id for Nmos

Plot 1D (pmos_03v3.lut)

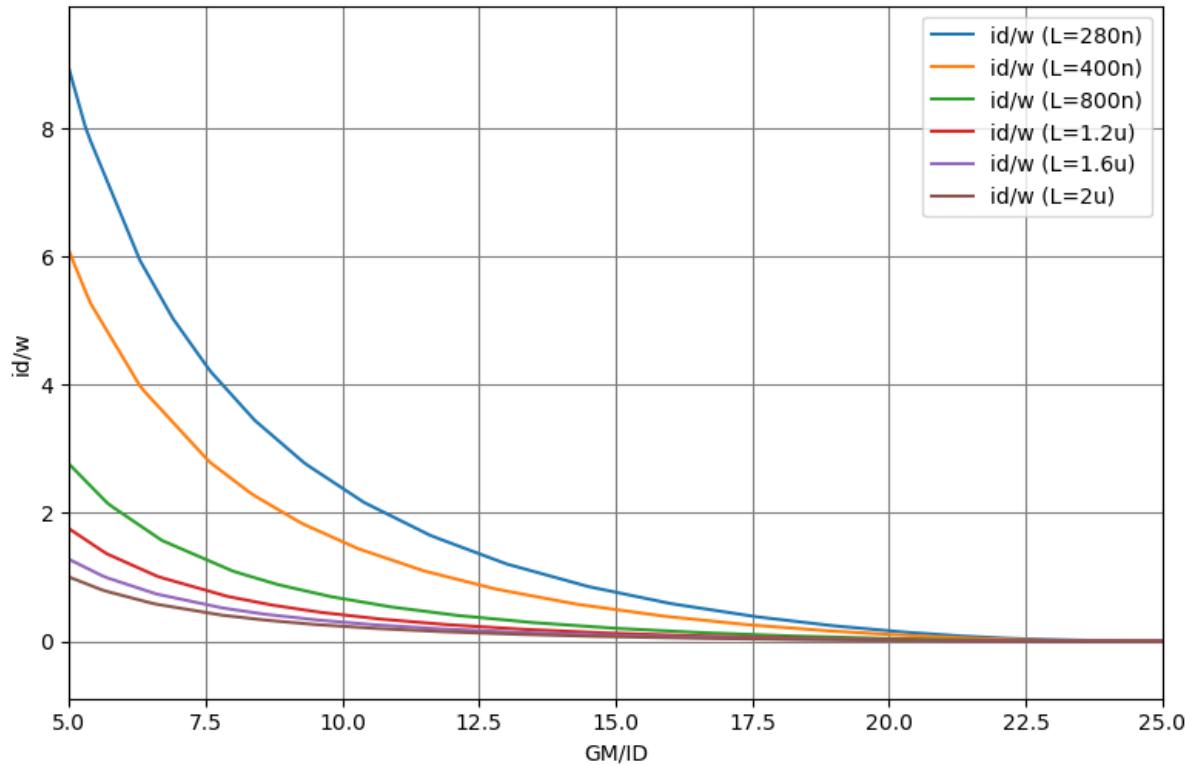


Figure 4 : : ID/W vs gm/id for Pmos

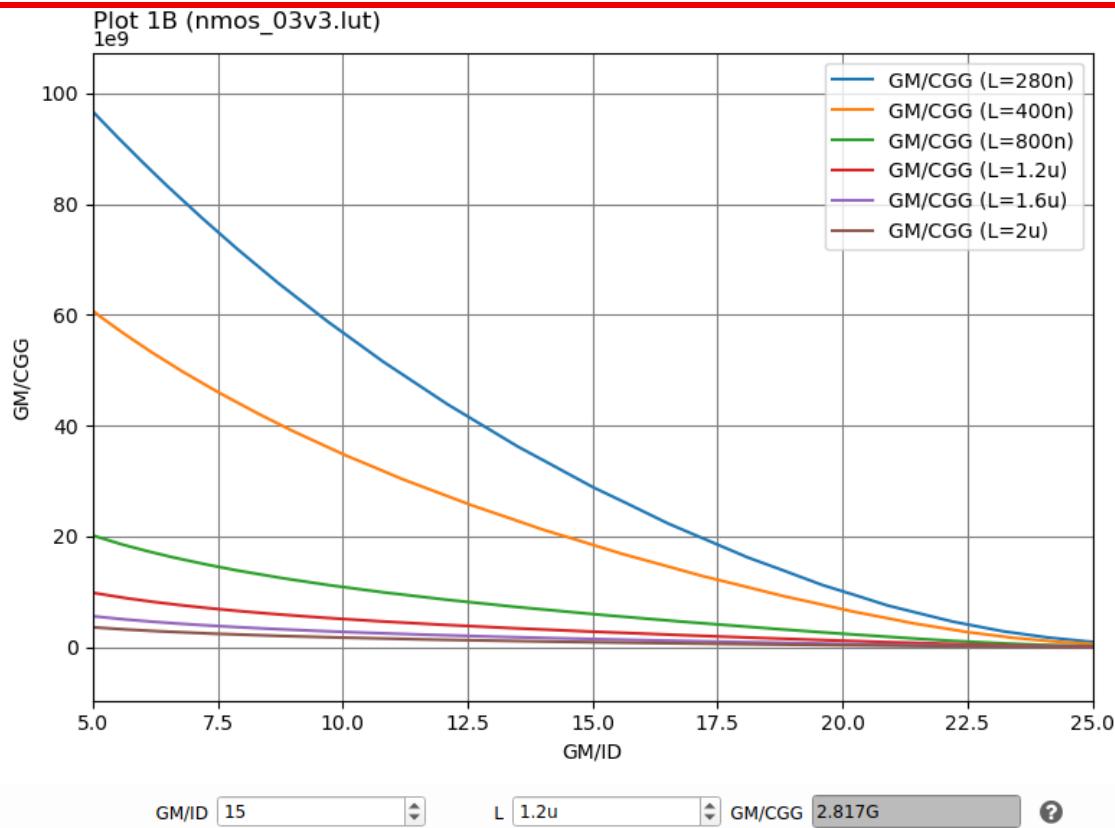


Figure 5 : gm/cgg vs gm/id for Nmos

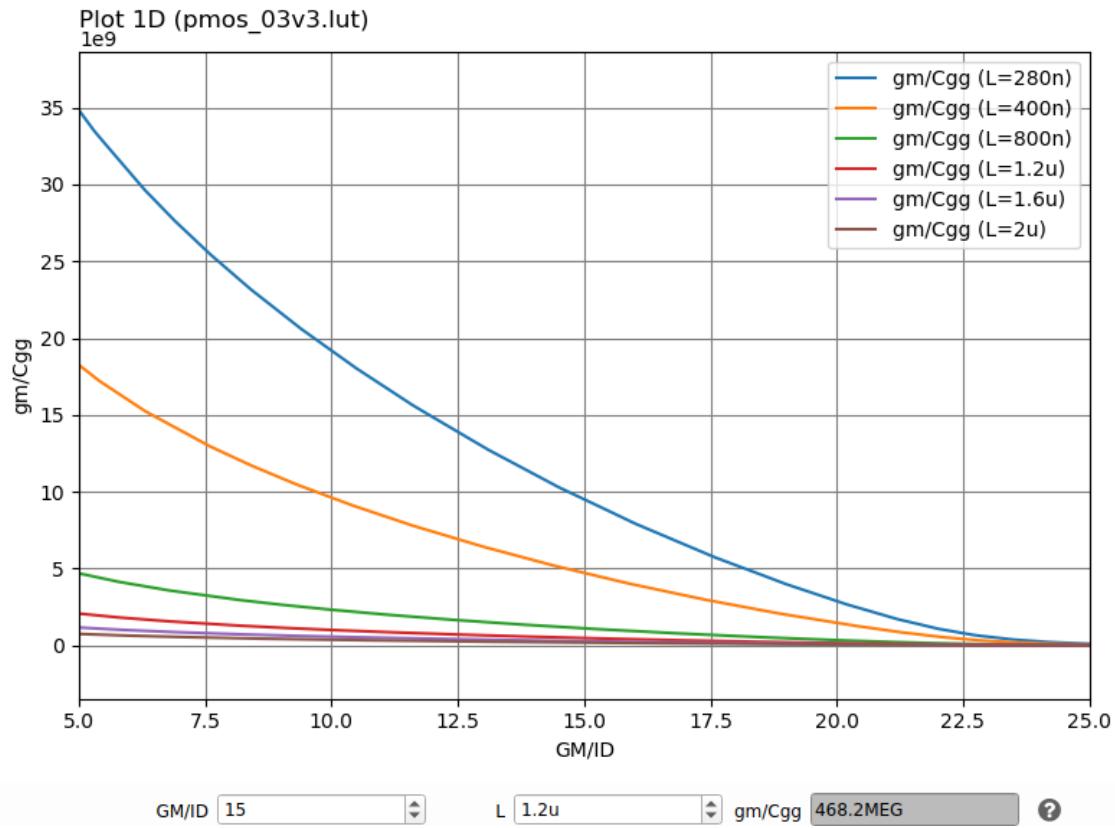
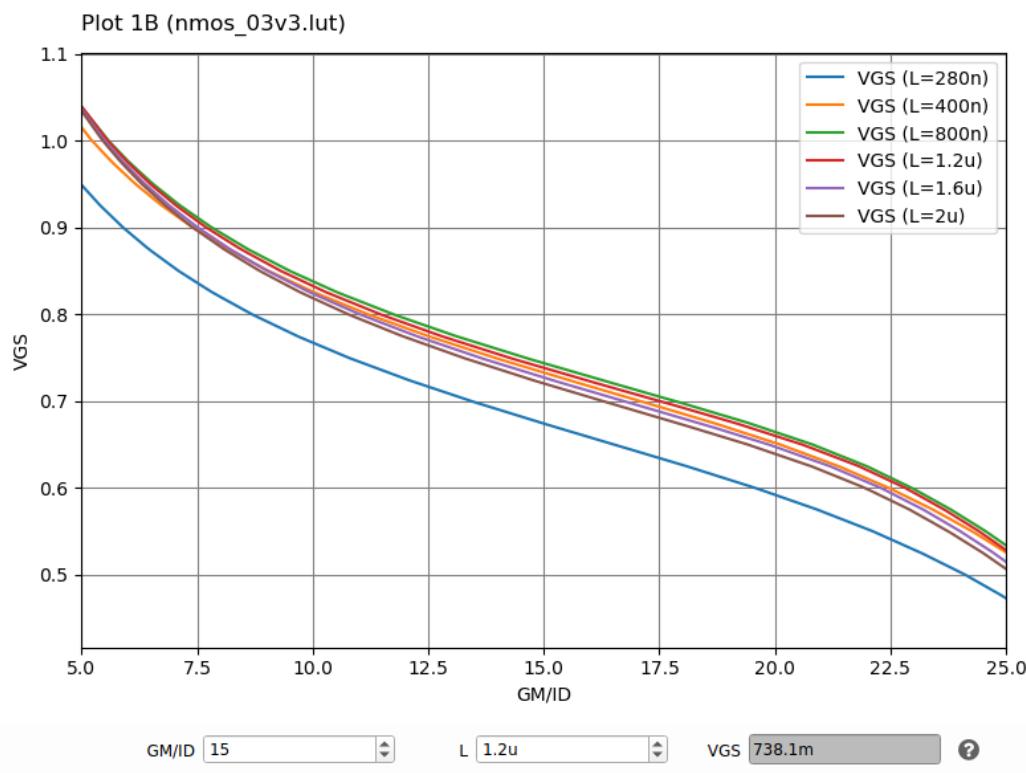
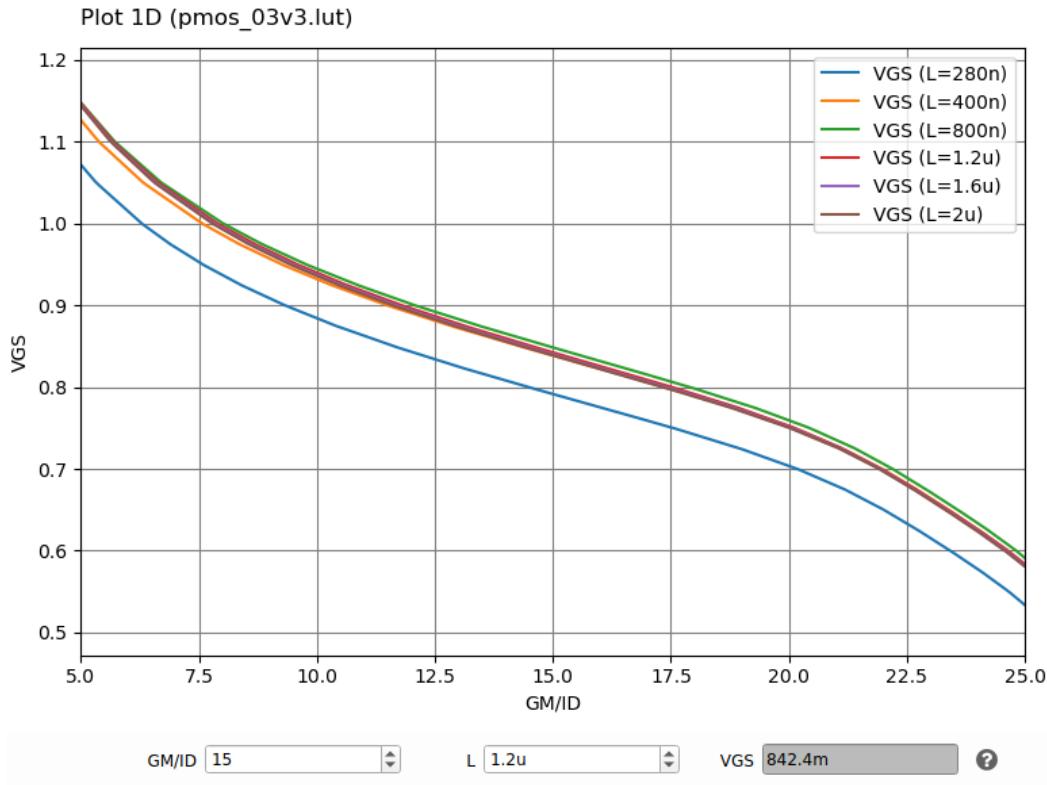


Figure 6 : gm/cgg vs gm/id for Pmos

Figure 7 : V_{GS} vs gm/id for NmosFigure 8 : V_{GS} vs gm/id for Pmos

Part 1: OTA Design

Parameter	0.18 μm
Supply voltage	1.8 V
Static gain error	$\leq 0.05\%$
CMRR @ DC	$\geq 74 \text{ dB}$
Phase margin (avoid pole-zero doublets)	$\geq 70^\circ$
OTA current consumption	$\leq 60 \mu\text{A}$
CMIR – high	$\geq 0.8 \text{ V}$
CMIR – low	$\leq 0.2 \text{ V}$
Output swing	0.2 – 1.6 V
Load	5 pF
Buffer closed loop rise time (10%–90%)	$\leq 70 \text{ ns}$
Slew rate (SR)	5 V/ μs

Use an ideal external 10uA DC current source in your test bench (not included in the OTA current consumption spec) but design your own bias circuit (current mirrors).

Create a schematic and an appropriate symbol for the OTA.

Report the following:

Detailed design procedure and hand analysis. Justify why you used NMOS or PMOS input pair for each stage

I chose a PMOS input pair since the CMIR is close to ground. Next, we proceed with the design and selection of device dimensions (W/L)

- $IB1 = 10\mu\text{A}$ and for stability we need **$IB2 = 4IB1 = 40\mu\text{A}$**
- Slow rate = $IB1/C_c = V/\mu\text{s} \rightarrow C_c = 2 \text{ pF}$
- We need OTA is to be used as a buffer then $\beta = 1$
- Static gain error = $0.05\% = 1/LG = 1/\beta A_{OL} = 1/A_{OL} \rightarrow A_{OL} = 2000 = 66.02 \text{ dB}$
- $T_{rise} = 2.2 \tau \rightarrow UGF = GBW = 1/2\pi\tau \rightarrow UGF = GBW = 5 \text{ MHz}$
- $UGF(\text{Hz}) = gm_1/2\pi C_c \rightarrow gm_1 = 62.86 \mu\text{s}$

Calculate gm/ID of the first stage.

$$(gm_1/Id_1)_{min} = 12.572 \text{ S/A}$$

Assign larger gain for the first stage

Assigning a larger first-stage gain reduces the impact of second-stage noise/offset when referred to the input and helps achieve the high DC gain needed for precision.

We will split the gain between the two stages. assume the first stage gain is twice that of the second stage (6dB difference).

$$Av_1 = 36\text{dB} = 63.3, Av_2 = 30\text{dB} = 31.623$$

Input Pair :

We will assume the Rout for input and load equal .

$$Av_1 \geq 63.3$$

$$gm/gds \geq 126.6$$

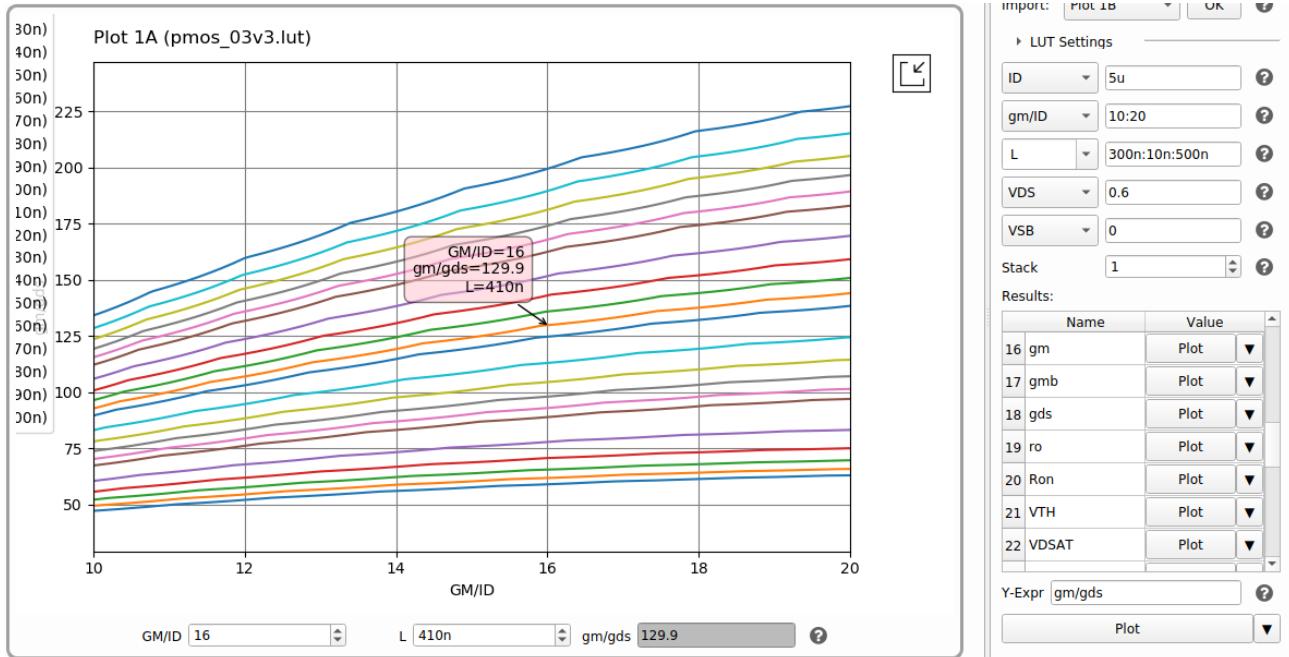


Figure 9 : gm/id vs gm/gds input pair

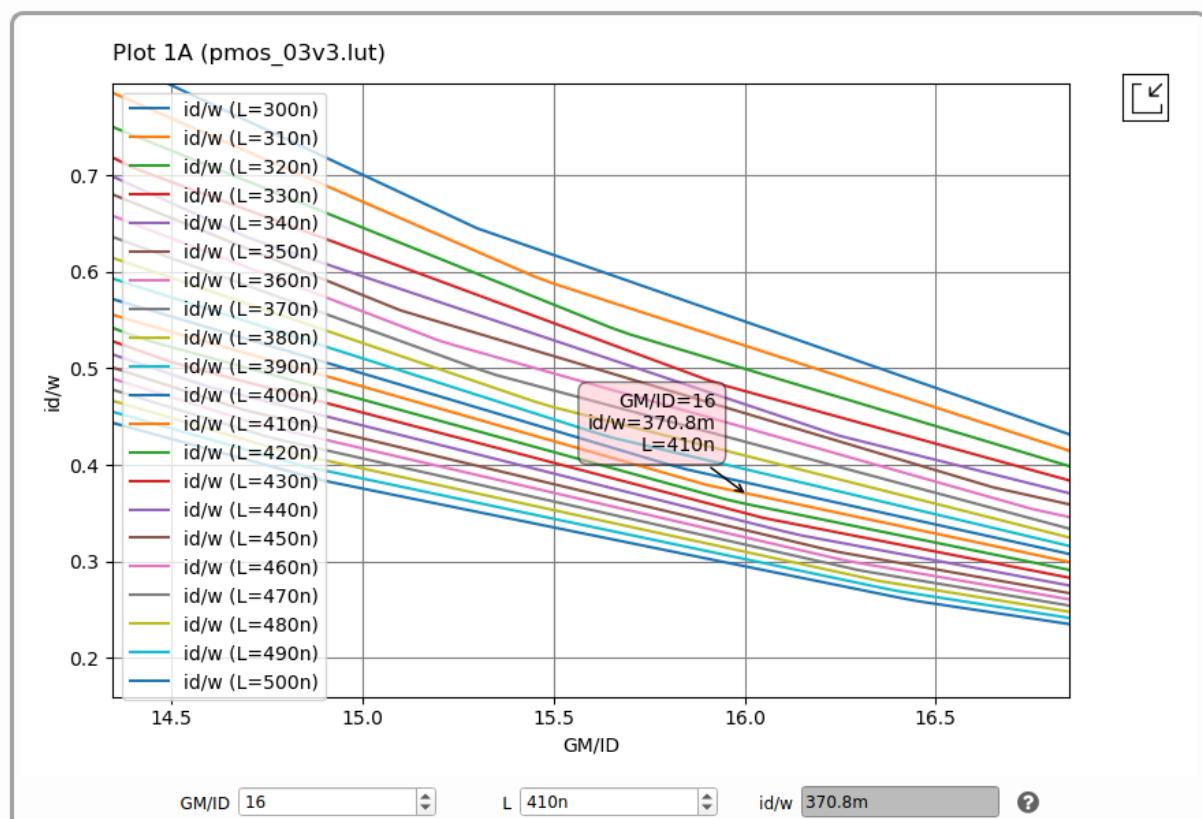


Figure 10 : gm/id vs id/w input pair

For input pair :

$$\text{gm/id} = 16, \text{gm/gds} = 129.9, L = 410 \text{ nm}, \text{id/w} = 370.8\text{m} \rightarrow W = 13.48 \text{ um}$$

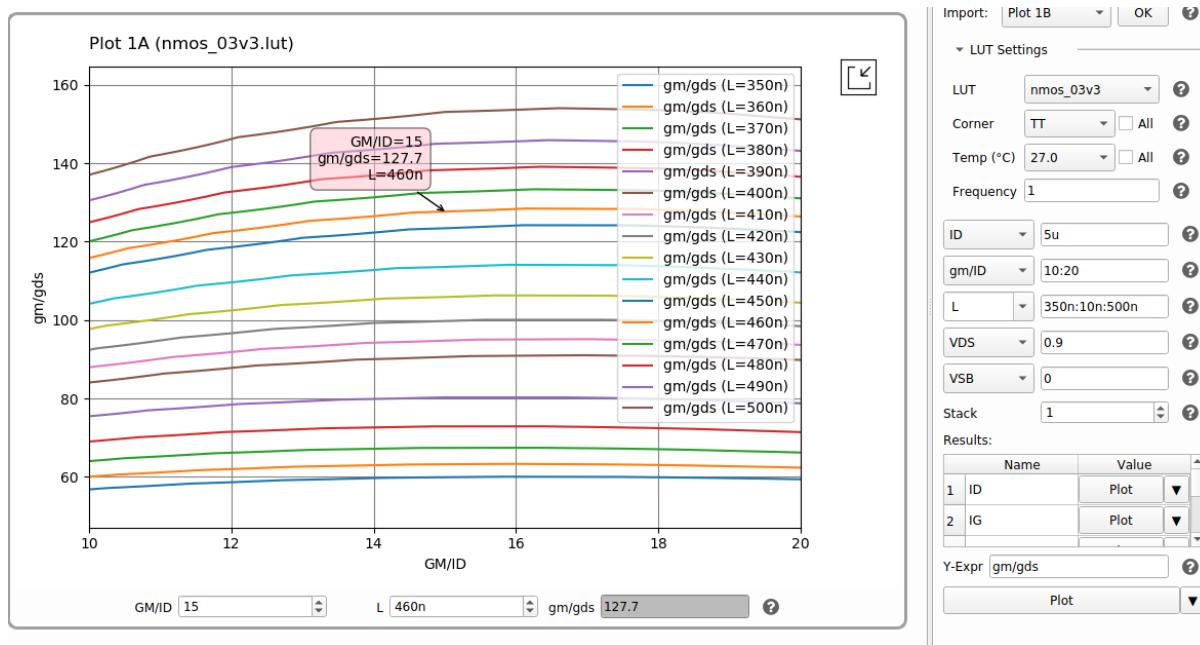
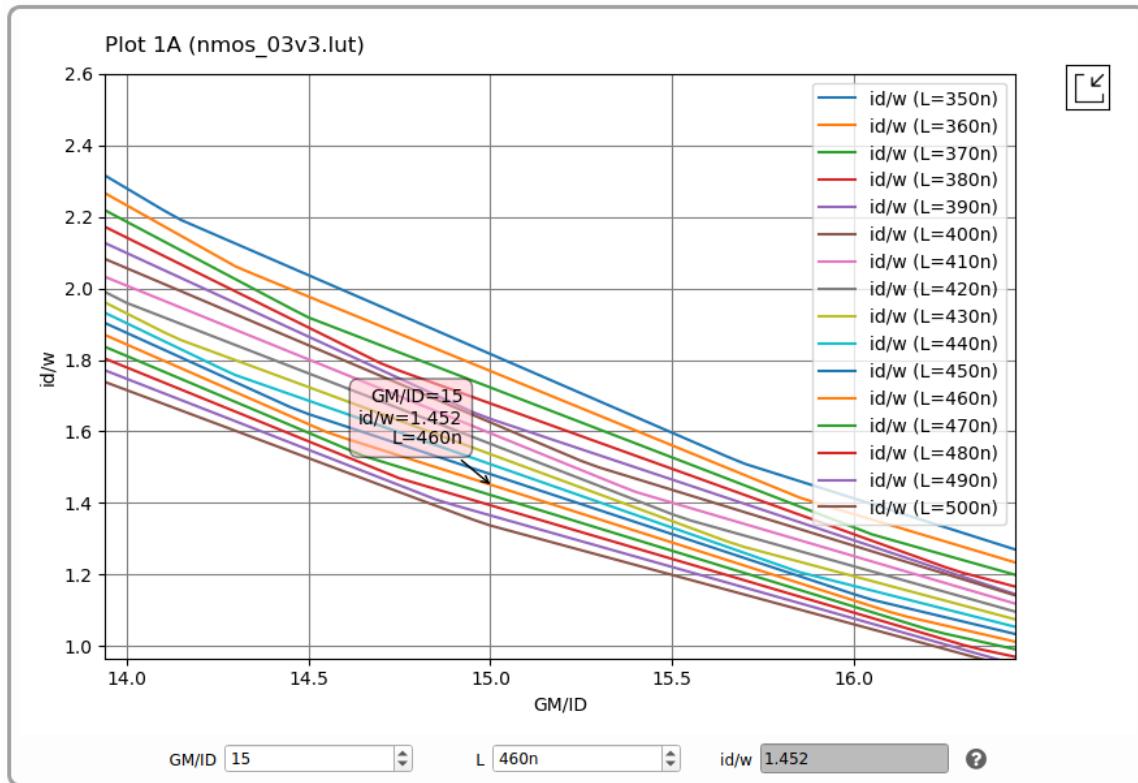
Results:

Name			TT-27.0
3	L	410n	
4	W	13.48u	
5	VGS	823.9m	
6	VDS	600m	
7	VSB	0	
8	gm/ID	15.94	
9	Vstar	125.4m	

Results:

Name			TT-27.0
16	gm	79.72u	
17	gmb	33.19u	
18	gds	614.2n	
19	ro	1.628MEG	
20	Ron	120k	
21	VTH	778.1m	
22	VDSAT	100.1m	

CM Load for 1st stage :

Figure 11 : gm/id vs gm/gds Nmos load for 1st stageFigure 12 : gm/id vs id/w Nmos load for 1st stage

For Nmos load 1st stage

$$gm/id = 15, gm/gds = 127.7, L = 460 \text{ nm}, id/w = 1.452 \rightarrow W = 3.44 \mu\text{m}$$

Results:

	Name	TT-27.0
3	L	460n
4	W	3.44u
5	VGS	744.8m
6	VDS	900m
7	VSB	0
8	gm/ID	14.8
9	Vstar	135.2m

Results:

	Name	TT-27.0
16	gm	73.98u
17	gmb	25.32u
18	gds	579.8n
19	ro	1.725MEG
20	Ron	180k
21	VTH	704.4m
22	VDSAT	108.2m

Tail Current mirror :

$$CMRR = Avd/Avcm = gm_1 * (ro_1//ro_3) * 2 * gm_3 * R_{ss}$$

$$\text{For } CMRR \geq 74 \text{ dB} = 5011.87 \rightarrow R_{ss} \geq 510.52 \text{ K}\Omega$$

$$\text{assume } gm/id = 15 \rightarrow gm = 150 \mu\text{S} \rightarrow gm/gds \geq 76.578 \rightarrow L = 500 \text{ n}$$

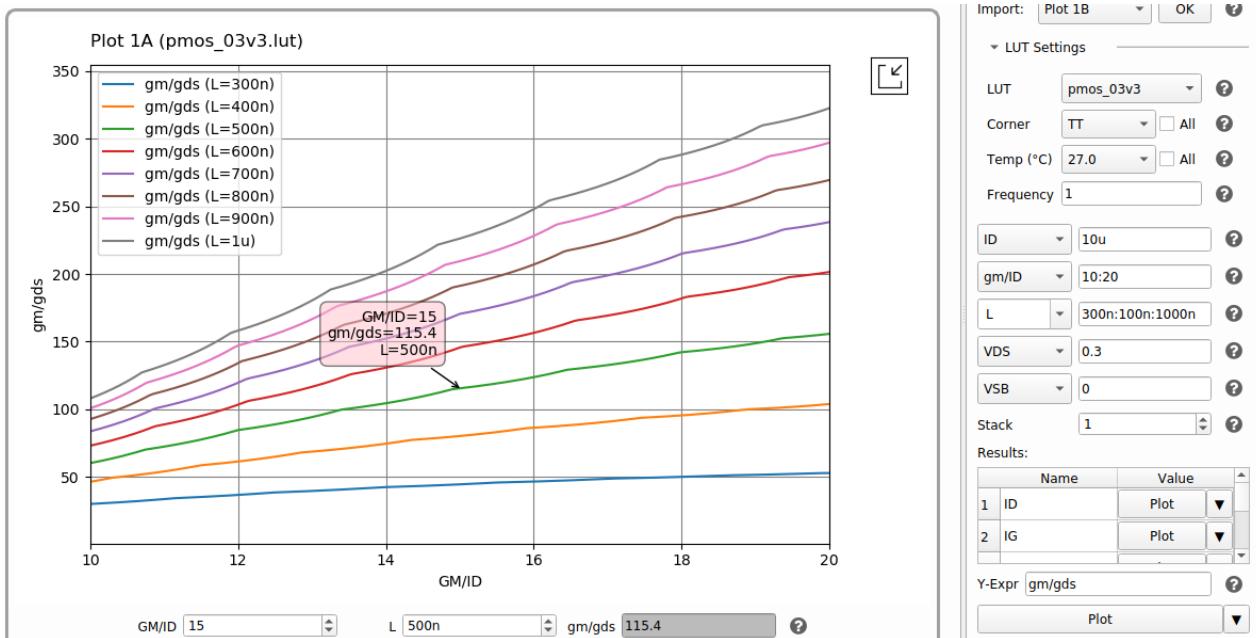


Figure 13 : gm/id vs gm/gds for tail CM

$$\text{CMIR}_{\text{high}} = \text{Vdd} - |\text{Vdsat}_5| - |\text{Vgs1}| > 0.8$$

$$|\text{Vdsat}_5| < 176.1 \text{ mV}$$

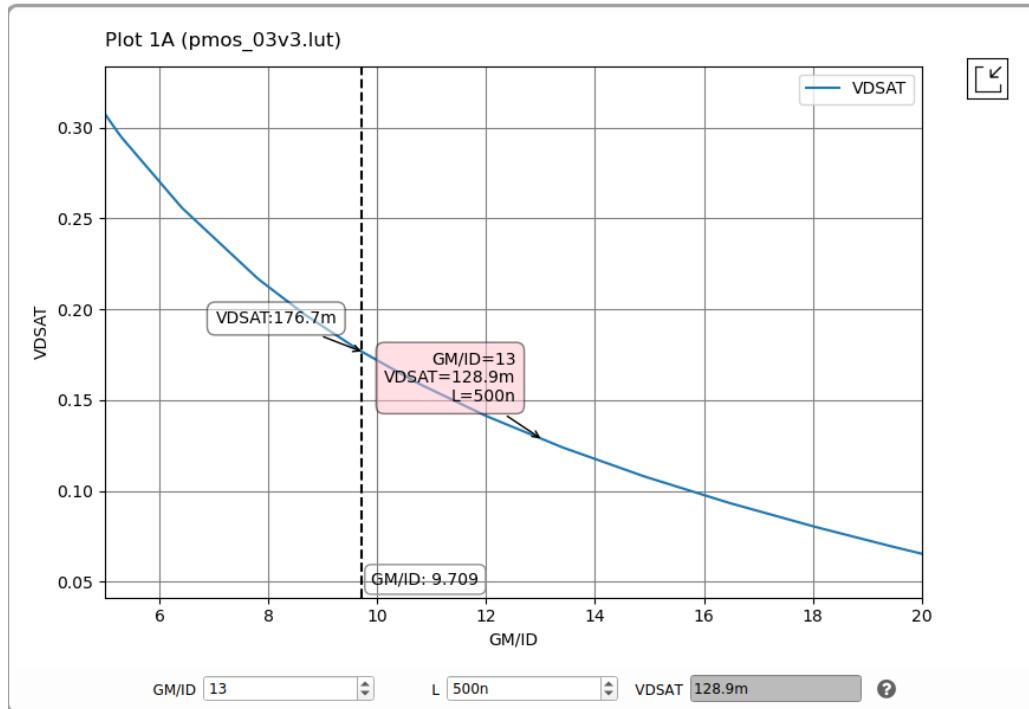


Figure 14 : gm/id vs V_{DSAT} for tail CM

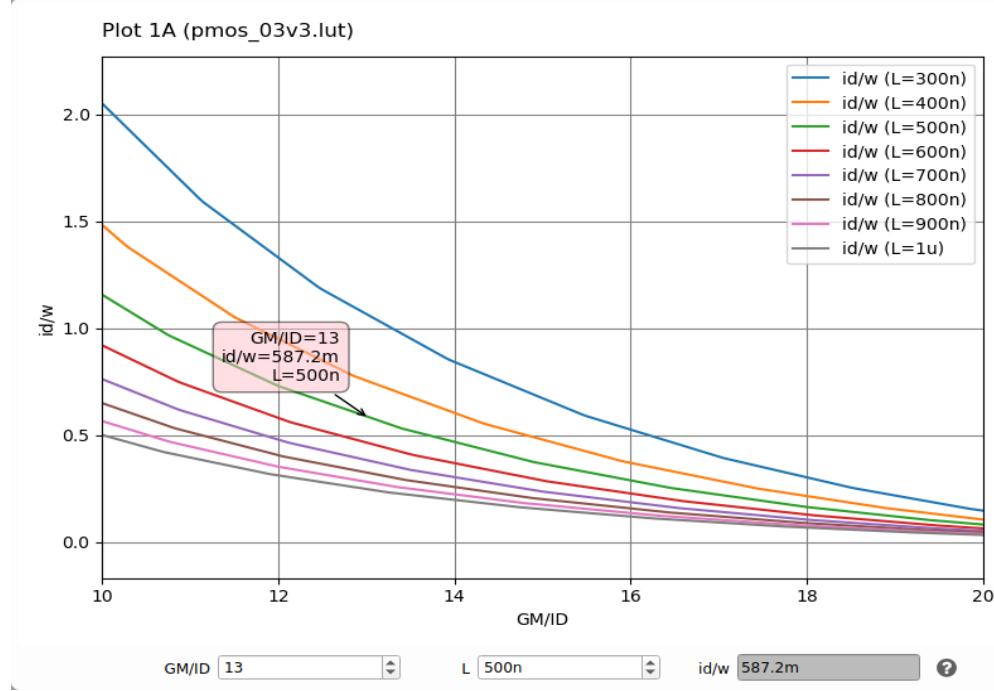


Figure 15 : gm/id vs $\frac{id}{w}$ For tail CM

For Tail Current Mirror :

$$gm/id = 13, L = 500n, w = 17.03u$$

Results:

	Name	TT-27.0
3	L	500n
4	W	17.03u
5	VGS	883.7m
6	VDS	300m
7	VSB	0
8	gm/ID	12.8
9	Vstar	156.3m

Results:

	Name	TT-27.0
16	gm	128u
17	gmb	56.97u
18	gds	1.376u
19	ro	726.5k
20	Ron	30k
21	VTH	786m
22	VDSAT	130.1m

Input of 2nd Stage :

$$\text{we want } wp_2 = 4wu \rightarrow wu = gm/Cc \rightarrow Gm_2/C_1 = 4Gm/Cc$$

$$C_1 = 5 \text{ pF}, C_c = 2 \text{ pF} = 0.4C_L, Gm_1 = 79.72 \text{ uS} \rightarrow Gm_2/Gm_1 = 10 \rightarrow Gm_2 = 797.2 \text{ uS}$$

For M7 tail CM , id = 40u

Results:

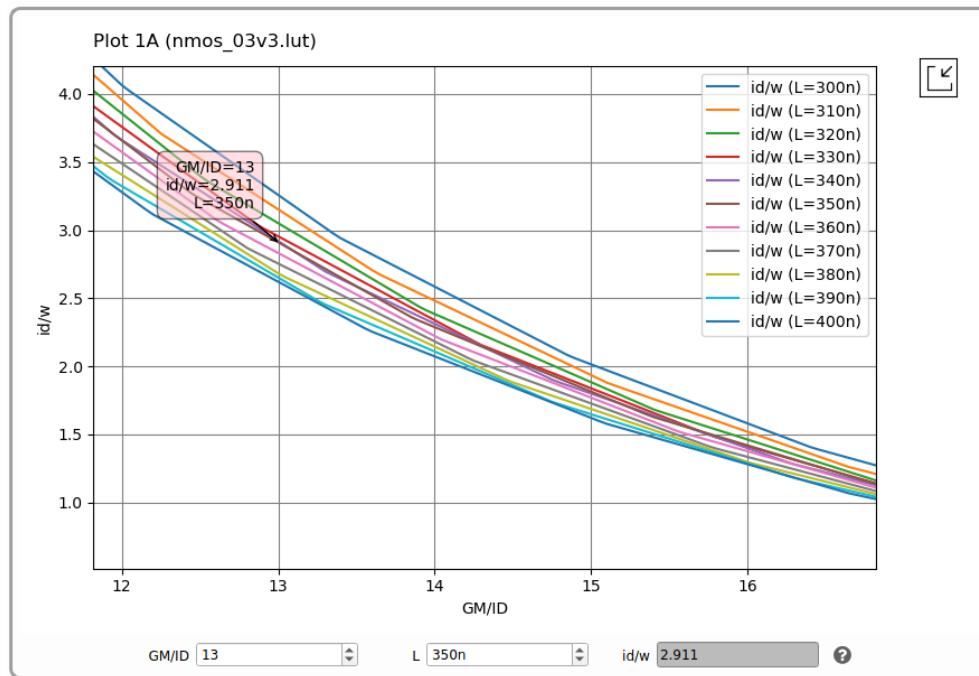
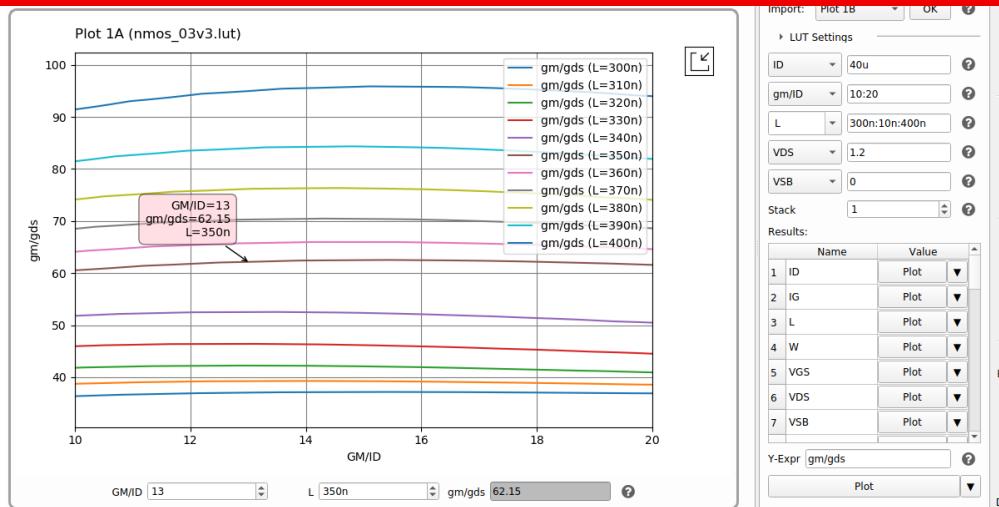
	Name	TT-27.0
3	L	500n
4	W	68.12u
5	VGS	883.7m
6	VDS	300m
7	VSB	0
8	gm/ID	12.8
9	Vstar	156.3m

Results:

	Name	TT-27.0
16	gm	512u
17	gmb	227.9u
18	gds	5.505u
19	ro	181.6k
20	Ron	7.5k
21	VTH	786m
22	VDSAT	130.1m

$$Av_2 = Av_{1/2} = 33.175 = gm_8 * (ro_7 // ro_8) \rightarrow Gm_2 = gm_8 \rightarrow ro_7 = 181.6 \text{ k}\Omega \rightarrow ro_8 = 54 \text{ K}\Omega$$

$$Gm_8/gds_8 \geq 43.048$$

**Results:**

Name		TT-27.0
16	gm	512u
17	gmb	145.8u
18	gds	8.242u
19	ro	121.3k
20	Ron	30k
21	VTH	675.7m
22	VDSAT	122.9m

Results:

Name		TT-27.0
3	L	350n
4	W	13.74u
5	VGS	742.1m
6	VDS	1.2
7	VSB	0
8	gm/ID	12.8
9	Vstar	156.3m

Component	W(μm)	L(nm)	gm(μS)	ID(μA)	gm/ID	Vdsat(mV)	Vov(mV)	V*(mV)
Input pair (PMOS)	13.48	410	79.72	5	16	100.1	45.8	125.4
CM load (NMOS)	3.44	460	73.98	5	15	108.2	40.4	135.2
Tail (PMOS)	17.03	500	128	10	13	130.1	97.7	156.3
2nd Stage NMOS	13.74	350	512	40	13	122.9	66.4	156.3

PART 3: Open-Loop OTA Simulation

Create a testbench similar to the one shown above (the shown schematic is from the 5T OTA lab). Note that IDC connection in the test bench (sourcing) may be different from the one shown above depending on the type of your input pair (PMOS).

Schematic :

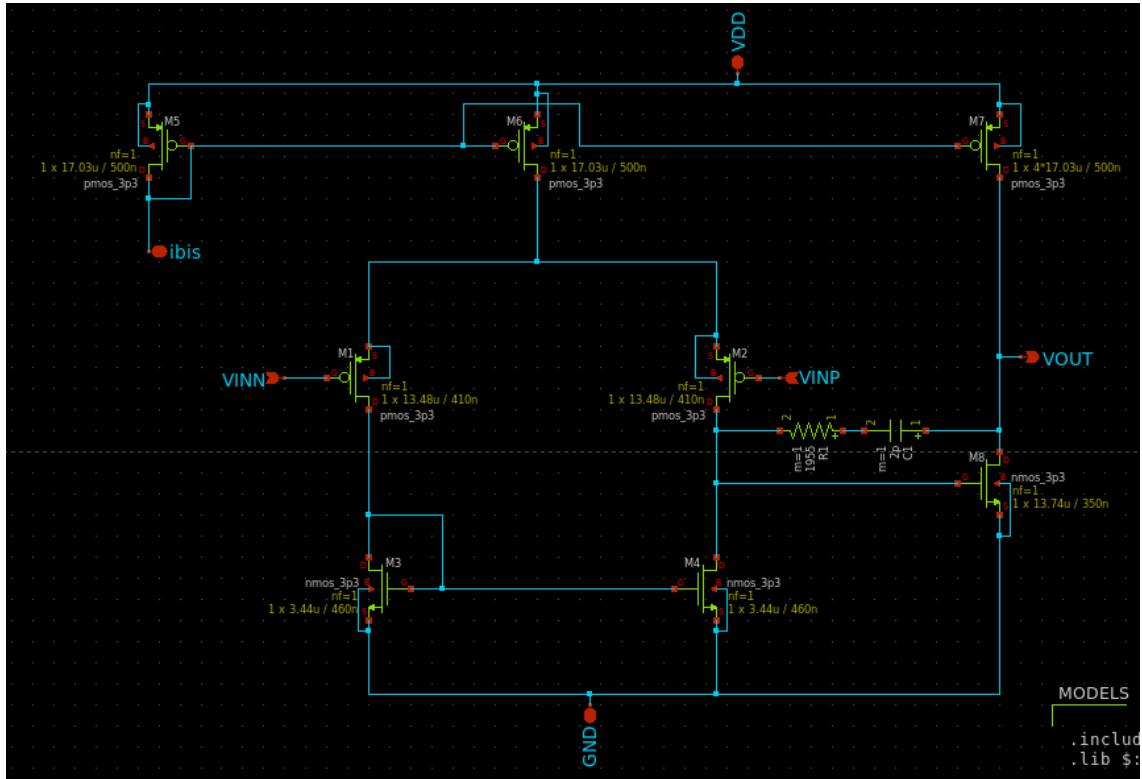


Figure 16 : Schematic of Two-Stage Miller OTA

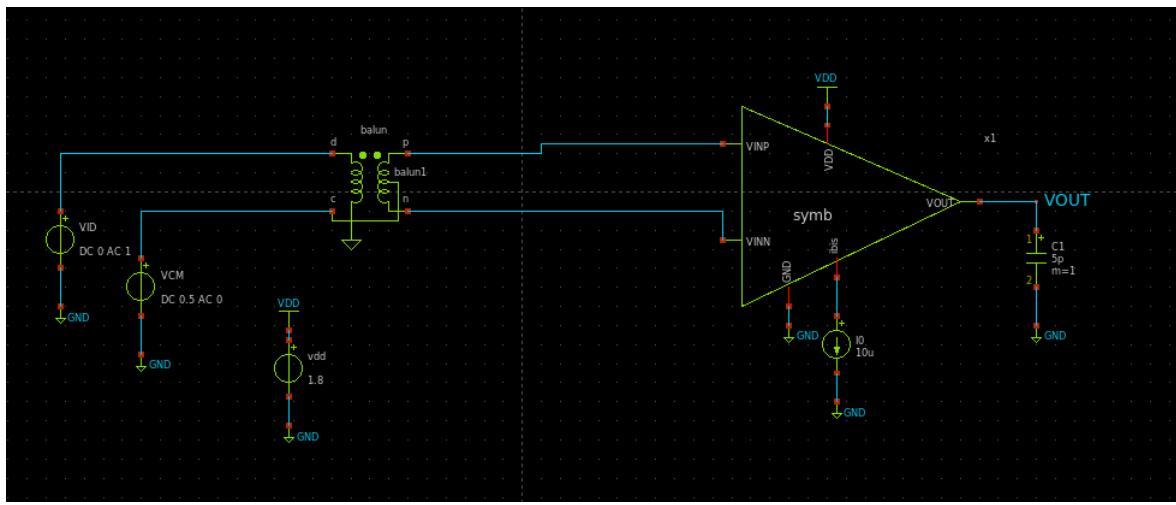


Figure 17 Testbench of Two-Stage Miller OTA with load

$-v_{thp} + v_{gs3} < V_{inCM} < V_{dd} - |V_{dsat_5}| - V_{gs1}$

$$143.3m < V_{inCM} < 853.9m$$

$$V_{inCM_middle} = 0.5V$$

device	m.x1.xm8.m0	m.x1.xm7.m0	m.x1.xm5.m0
model	nmos_3p3.12	pmos_3p3.12	pmos_3p3.12
id	3.98478e-05	3.98478e-05	1e-05
gm	0.000514809	0.000526808	0.000132166
gds	8.51075e-06	2.47467e-06	6.01365e-07
vgs	0.743218	0.880594	0.880594
vth	0.678577	0.788382	0.788339
vds	0.967327	0.832664	0.880593
vdsat	0.121977	0.125716	0.125745

BSIM4v5: Berkeley Short Channel IGFET Model-4			
device	m.x1.xm6.m0	m.x1.xm1.m0	m.x1.xm2.m0
model	pmos_3p3.12	pmos_3p3.12	pmos_3p3.12
id	9.66089e-06	4.83045e-06	4.83045e-06
gm	0.000127939	7.8162e-05	7.8162e-05
gds	9.44713e-07	5.667e-07	5.667e-07
vgs	0.880594	0.824203	0.824203
vth	0.788761	0.781709	0.781709
vds	0.415795	0.640981	0.640981
vdsat	0.125462	0.0978128	0.0978128

BSIM4v5: Berkeley Short Channel IGFET Model-4			
device	m.x1.xm3.m0	m.x1.xm4.m0	m.x1.xm2.m0
model	nmos_3p3.8	nmos_3p3.8	nmos_3p3.8
id	4.83045e-06	4.83045e-06	4.83045e-06
gm	7.29607e-05	7.29607e-05	7.29607e-05
gds	5.97758e-07	5.97758e-07	5.97758e-07
vgs	0.743219	0.743219	0.743219
vth	0.704146	0.704146	0.704146
vds	0.743218	0.743218	0.743218
vdsat	0.107168	0.107168	0.107168

Figure 18 : OP Annotation

ALL transistor in saturation $V_{ds} > V_{dsat}$

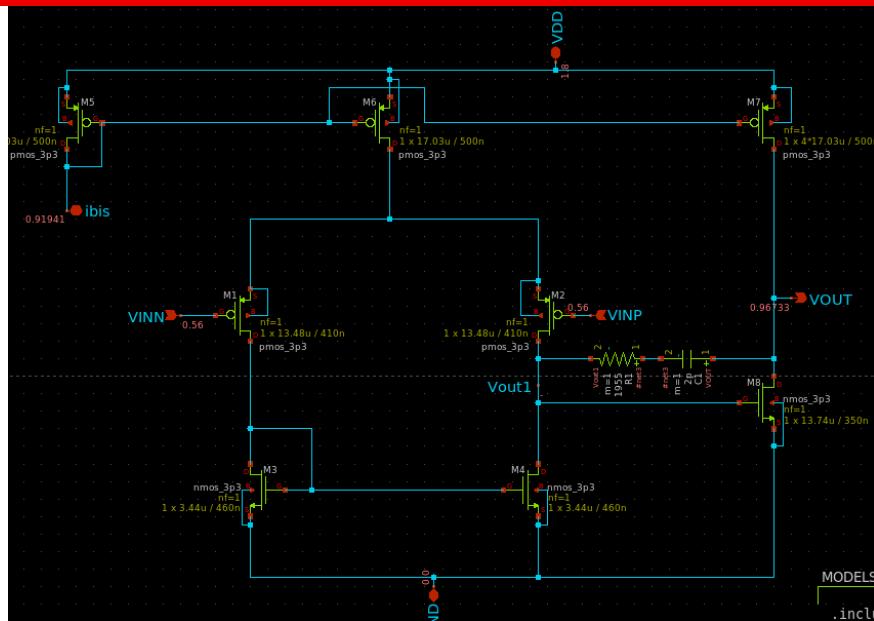


Figure 19 : DC node Voltage

```
x1_vout1 = 7.432212e-01
```

- Is the current (and gm) in the input pair exactly equal?

Yes

- What is DC voltage at the output of the first stage? Why?

$V_{out_1} = 0.7432V = V_{ds\ 4}$, because the V_{out_1} follow the mirror node V_f

- What is DC voltage at the output of the second stage? Why?

$V_{out} = 0.967$, because we need the maximum Swing so the V_{out} must equal $V_{dd}/2$

Diff small signal ccs:

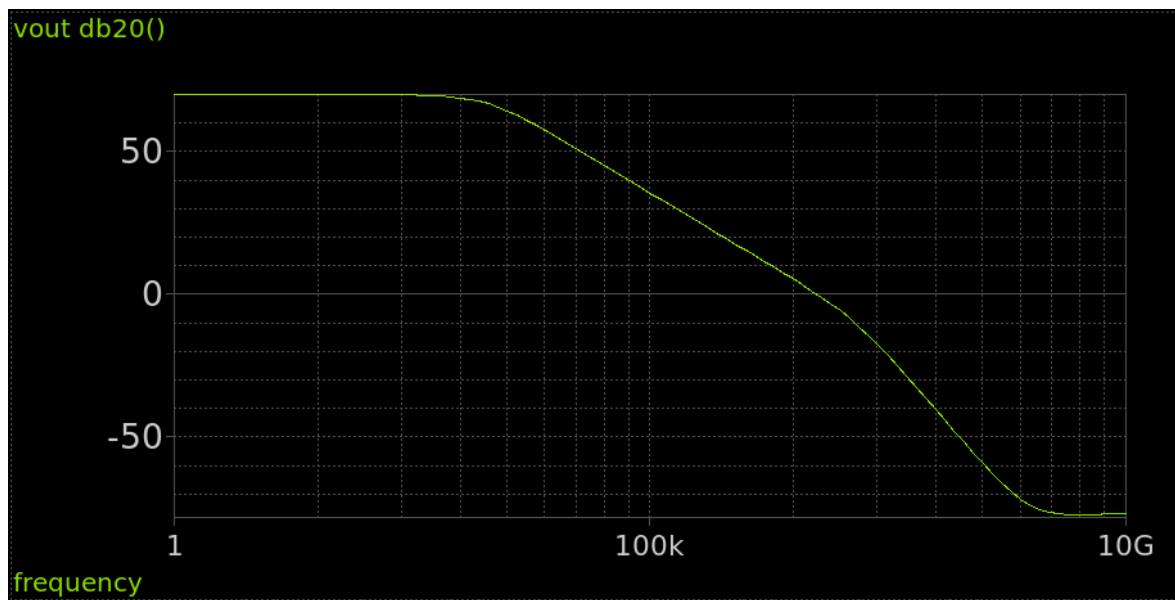


Figure 20 : diff gain (in dB) vs frequency

```
No. of Data Rows : 101
gain              = 3.132729e+03 at= 1.000000e+00
bw                = 1.913221e+03
gbw               = 5.720885e+06
binary raw file "lab9_ac.raw"
```

Figure 21 : gain & BW & GBW from simulation

Analytical :

DC gain = $gm_1,2 \cdot ro_1 // ro_3 \cdot gm_8 \cdot ro_7 // ro_8$ from OP annotation = 3148.9 = 69.96 dB

Bw = $1 / 2\pi \cdot Rout_1 \cdot Cc \cdot Av_2 = 1.98 \text{ kHz}$

GBW = DC gain * Bw = 6.23 kHz

Parameter	Hand Calculation (Analytical)	Simulation Result
DC Gain	3148.9 (69.96 dB)	3132.7 (69.91 dB)
Bandwidth (BW)	1.98 kHz	1.91 kHz
GBW	6.23 MHz	5.72 MHz

CM small signal ccs

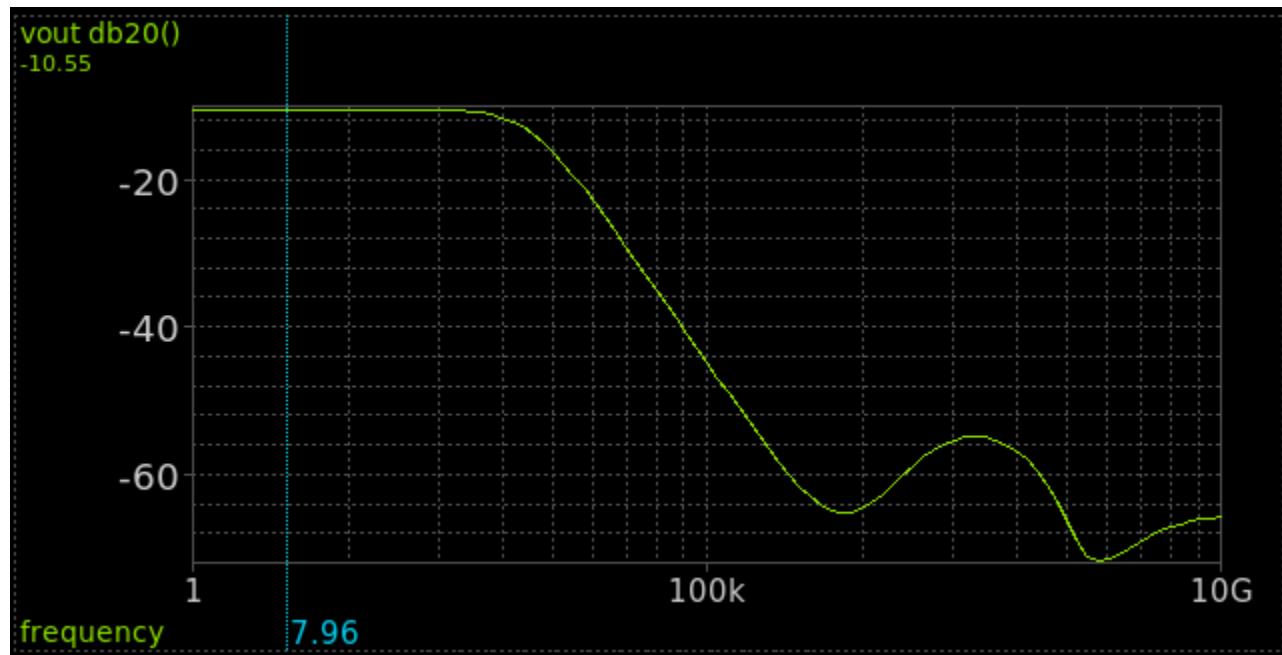


Figure 22 : CM gain in dB vs frequency.

```
cmgain = 2.969728e-01 at= 1.000000e+00
```

Analytical:

$$Av_{CM} = \frac{1}{2} \cdot gm_3 \cdot 4 \cdot r_o6 \cdot Av_2(gm_8 \cdot r_o8 // r_o7)$$

$$R_{SS} = r_o6$$

$$Av_{CM} = 0.3035$$

	Hand Calculation (Analytical)	Simulation Result
Av _{CM}	0.3035 (-10.35dB)	0.2969 (-10.55dB)

(Optional) CMRR:

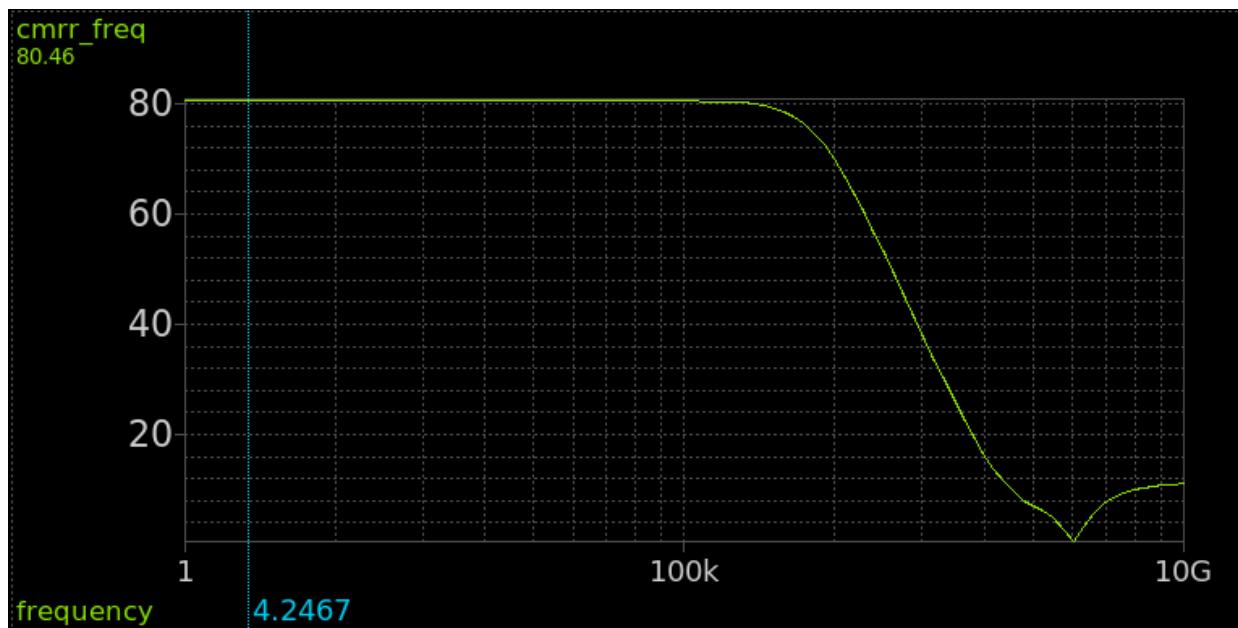


Figure 23 :: CMRR in dB vs frequency at V_{ICM} at the middle of the CMIR

Analytical

$$\text{CMRR} = \text{Avd} / \text{Acm} = 3148.9 / 0.3035 = 10375.2883 = 80.3\text{dB}$$

	Hand Calculation (Analytical)	Simulation Result
CMRR	80.3dB	80.46dB

(Optional) Diff large signal:

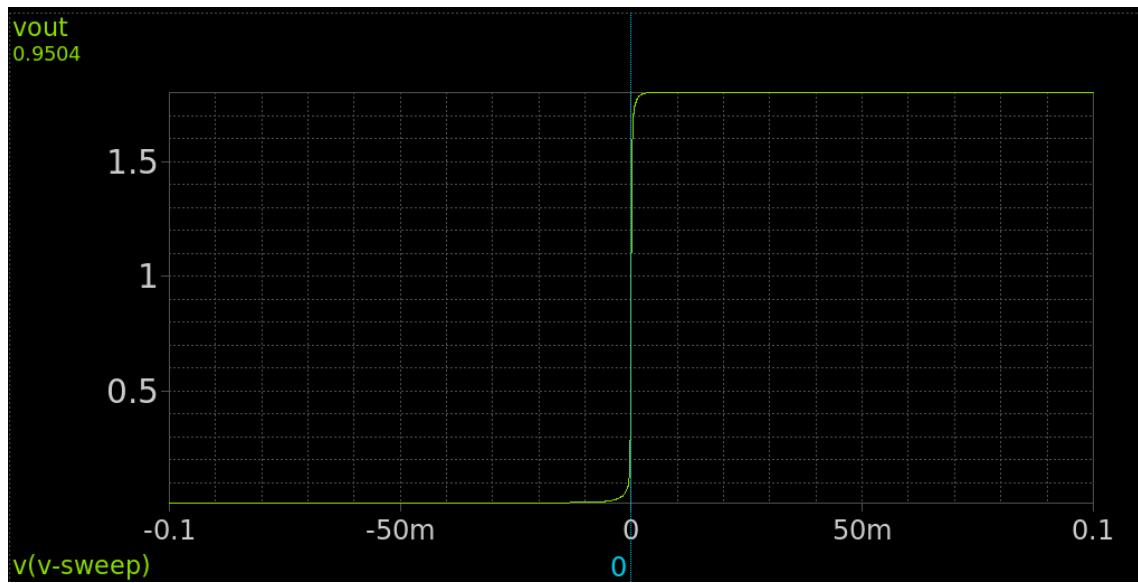


Figure 24 : V_{OUT} vs VID Diff large signal

- From the plot, what is the value of V_{OUT} at $VID = 0$. Compare it with the value you obtained in DC OP.

$V_{OUT} = 0.9504V$

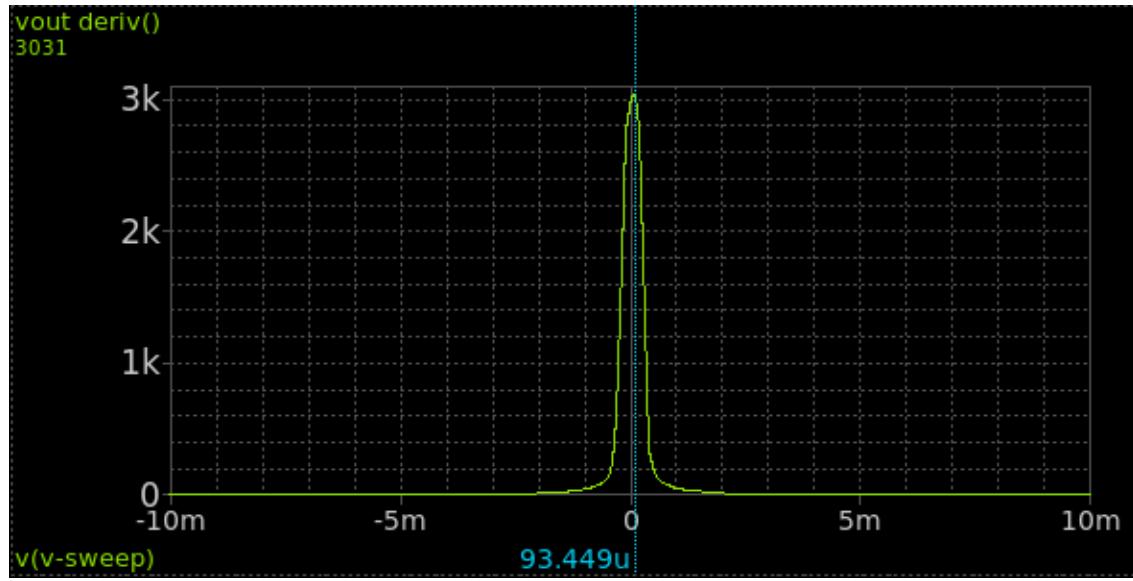


Figure 25 : the derivative of V_{OUT} vs VID .

The peak of curve is 3031 and the Avd = 3132.7 is very close to Avd

The derivative of V_{OUT} vs VID peaks at the same value as Avd because mathematically the slope of the transfer curve at the operating point is the small-signal gain.

CM large signal ccs (region vs VICM):

Plot “region” OP parameter vs VICM for the input pair and the tail current source (0 cut-off, 1 triode, 2 sat, 3 subth, and 4 breakdown)

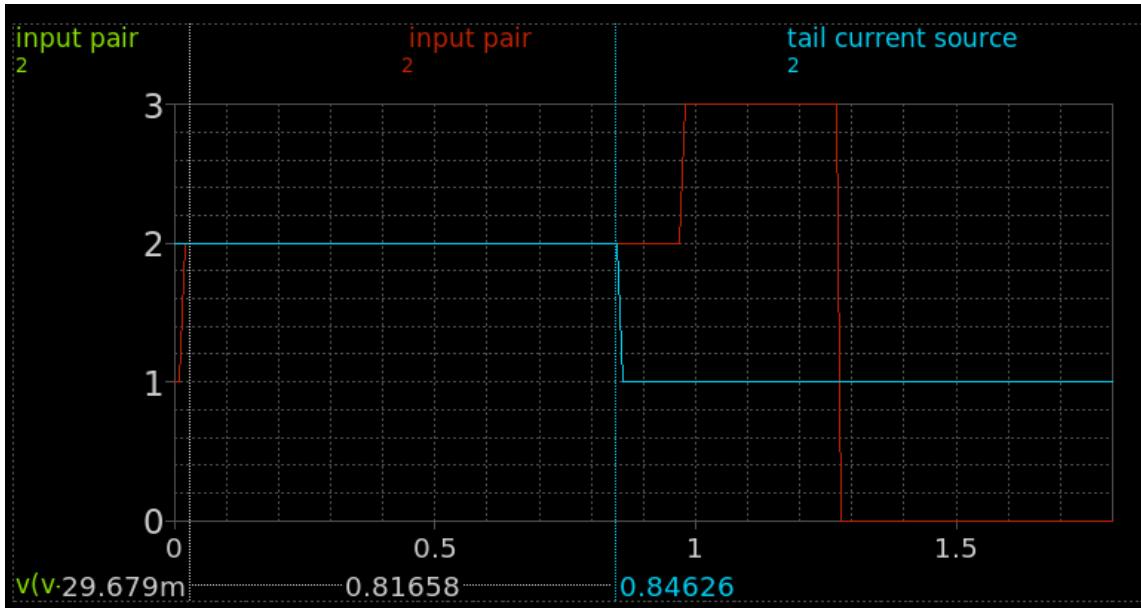


Figure 26 : region OP parameter vs VICM

From the plot: CMIR-Low = 0.0297 V, CMIR-high = 0.8463 V

From DC OP:

$$\text{CMIR-Low} = -VGS1,2 + Vdsat1,2 + VGS3,4 = -0.0385 \text{ V}$$

$$\text{CMIR-high} = Vdd - Vdsat5 - VGS1,2 = 0.850 \text{ V}$$

	Hand Calculation (Analytical)	Simulation Result
CMIR-high	0.850 V	0.8463 V
CMIR-Low	- 0.0385 V	0.0297 V

(Optional) CM large signal ccs (GBW vs VICM) :

Plot GBW vs VICM. Plot the results overlaid on the results of the previous method (region parameter).

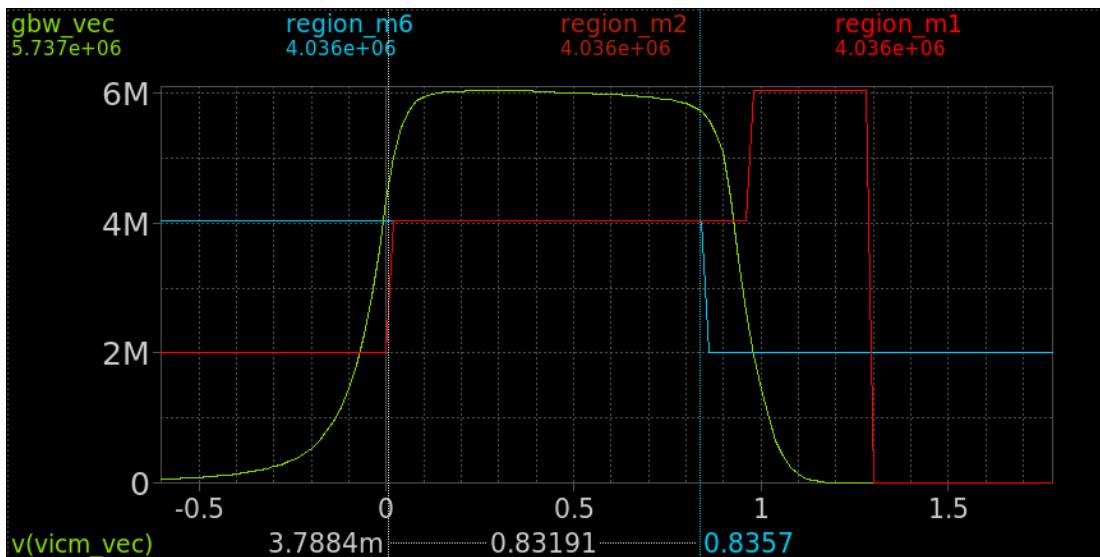


Figure 27 : GBW & region OP parameter vs VICM overlaid

Annotate the CM input range. Calculate the input range as the range over which the GBW is within 90% of the max GBW, i.e., 10% reduction in GBW.

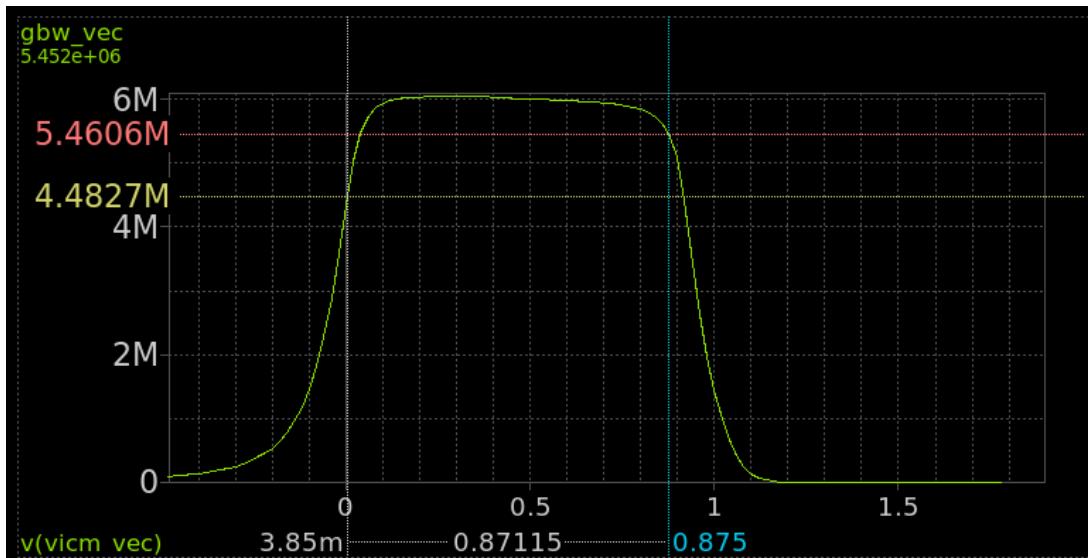


Figure 28 : GBW vs VICM Annotate the CM input range

```
No. of Data Rows : 101
gain           = 1.679918e-04 at= 1.258925e+08
bw             = 1.667128e+07
max_gbw        = 6.054206e+06 at= 2.511886e+04
max_vicm       = 8.753079e-01
min_vicm       = 3.856109e-02
cmir = 8.367468e-01
```

Figure 29 : gain & BW & max_gbw & max&min Vicm &Vicm_range

PART 4: Closed-Loop OTA Simulation :

Create a new testbench with the OTA connected in a unity gain buffer feedback configuration (the shown schematic is from the 5T OTA lab). Place a current probe (iprobe) or a zero voltage source in the feedback loop.

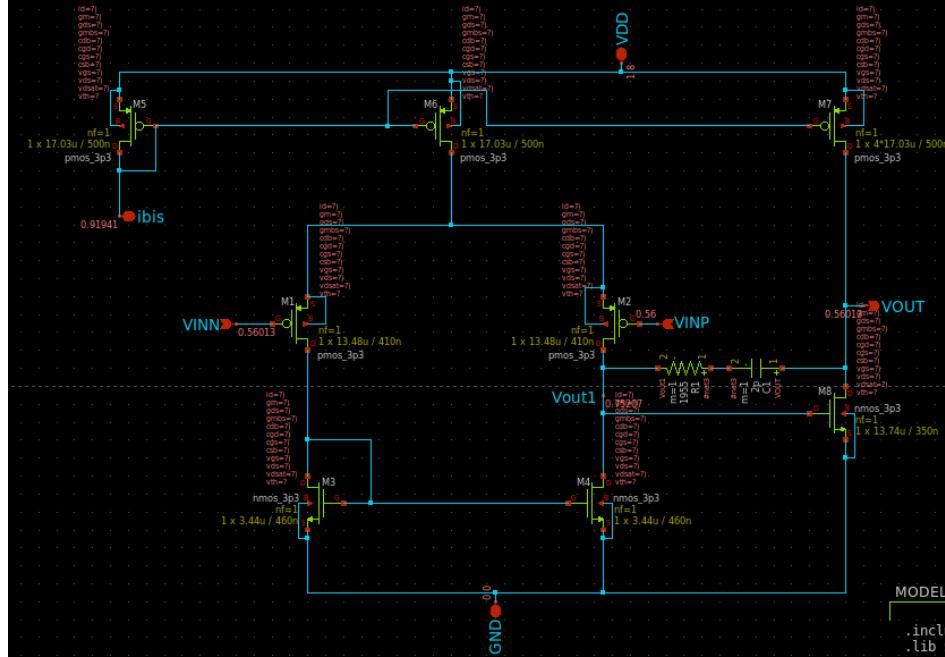


Figure 30 : Schematic of 5T-OTA with DC OP point clearly annotated

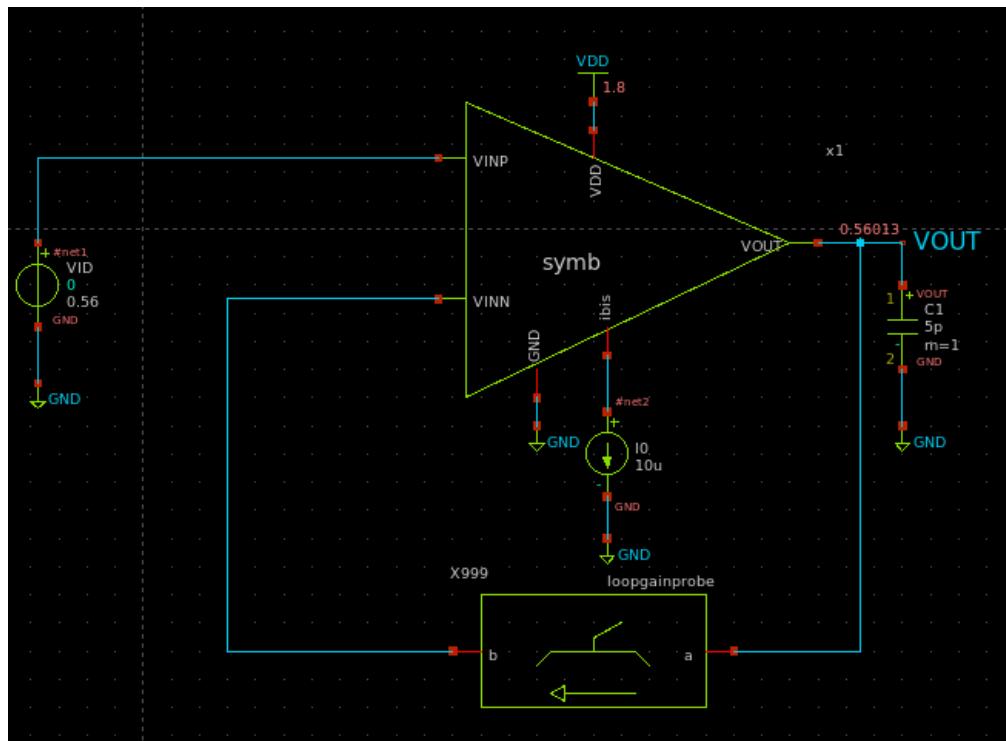


Figure 31 : testbench Closed-Loop OTA with DC OP point clearly annotated

BSIM4v5: Berkeley Short Channel IGFET Model-4			
device	m.x1.xm8.m0	m.x1.xm7.m0	m.x1.xm5.m0
model	nmos_3p3.12	pmos_3p3.12	pmos_3p3.12
id	4.07546e-05	4.07546e-05	1e-05
gm	0.000520237	0.000537349	0.000132166
gds	9.81766e-06	2.03236e-06	6.01365e-07
vgs	0.752069	0.880594	0.880594
vth	0.681985	0.788012	0.788339
vds	0.560125	1.23987	0.880593
vdsat	0.125249	0.125964	0.125745

BSIM4v5: Berkeley Short Channel IGFET Model-4			
device	m.x1.xm6.m0	m.x1.xm1.m0	m.x1.xm2.m0
model	pmos_3p3.12	pmos_3p3.12	pmos_3p3.12
id	9.6608e-06	4.82775e-06	4.83305e-06
gm	0.000127938	7.81288e-05	7.81837e-05
gds	9.44879e-07	5.66367e-07	5.70342e-07
vgs	0.880594	0.824168	0.8243
vth	0.788761	0.781708	0.781731
vds	0.415698	0.641115	0.632227
vdsat	0.125462	0.0977928	0.0978553

BSIM4v5: Berkeley Short Channel IGFET Model-4			
device	m.x1.xm3.m0	m.x1.xm4.m0	m.x1.xm2.m0
model	nmos_3p3.8	nmos_3p3.8	nmos_3p3.8
id	4.82775e-06	4.83305e-06	4.83305e-06
gm	7.29312e-05	7.29929e-05	7.29929e-05
gds	5.97511e-07	5.95633e-07	5.95633e-07
vgs	0.743183	0.743183	0.743183
vth	0.704146	0.704128	0.704128
vds	0.743181	0.752069	0.752069
vdsat	0.107147	0.107157	0.107157

: OP Annulation Closed-Loop OTA

Are the DC voltages at the input terminals of the op-amp exactly equal? Why?

No, because the mismatch of negative feedback and Aol is Not infinite

Is the DC voltage at the output of the first stage exactly equal to the value in the open-loop simulation? Why?

x1.vout1 = 7.520726e-01

No, because the mismatch of negative feedback and Aol is Not infinite

Is the current (and gm) in the input pair exactly equal? Why?

No, because the mismatch of negative feedback and Aol is Not infinite

Loop gain:

- Plot loop gain in dB and phase vs frequency.

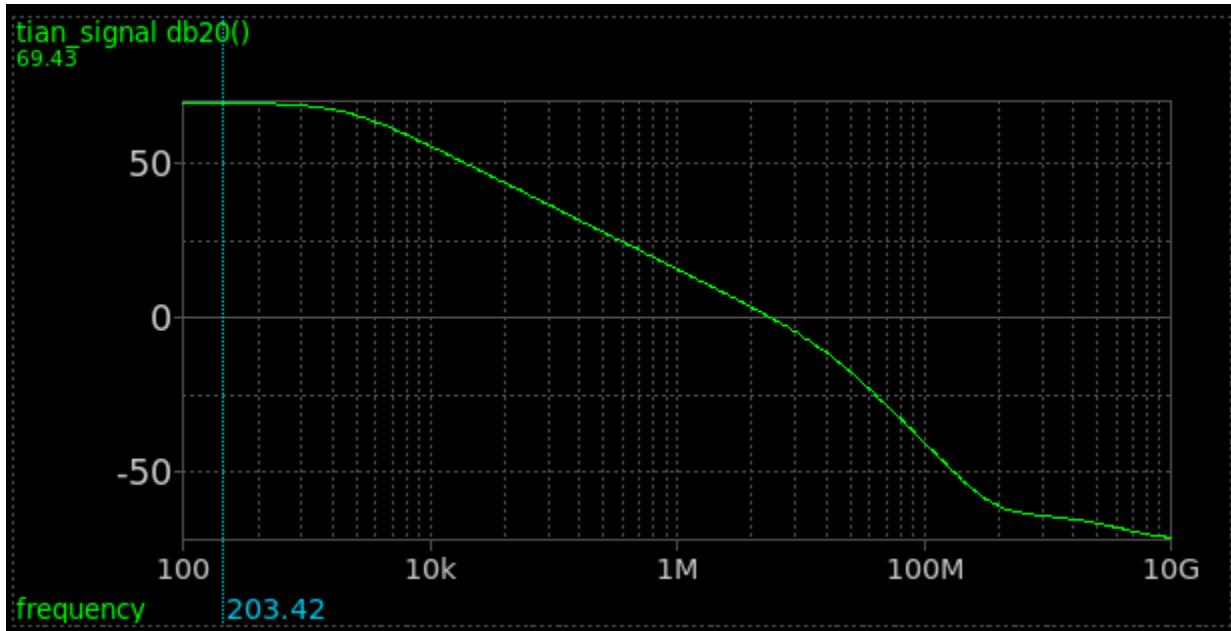


Figure 32 : loop gain in dB and phase vs frequency.

```
No. of Data Rows : 401
dc_gain_db          = 6.946466e+01
dc_gain_lin         = 2.973260e+03
bw                  = 2.054627e+03
gbw = 6.108940e+06
fu                  = 5.771949e+06
pm                  = 7.026225e+01
```

Figure 33 : Dc gain & BW & GBW & fu & PM from simulation

Parameter	Open-Loop Simulation	STB Simulation (Closed loop)
DC Gain	3132.7 (69.91 dB)	2973.3 (69.46 dB)
fu (Unity-Gain Freq.)	5.72 MHz	5.77 MHz
GBW (Gain-Bandwidth)	5.72 MHz	6.10 MHz

Comment :

The results are approximately equal due to the buffer connection as $LG = \beta * AOL$ and $\beta \approx 1$ so $LG = AOL$

Analytical :

$$W_u = g_m 1 / C_c = 39.064 \text{ Mrad/sec}$$

$$W_p = g_m 8 / C_l = 104.05 \text{ Mrad/sec}$$

$$PM = 90 - \tan^{-1}(W_u/W_p) = 69.4$$

Comment :

The results are approximately the same.

Parameter	Hand Calculation (Analytical)	Simulation Result
PM	69.4	70.26

Slew rate:

- Report Vin and Vout overlaid. , "PULSE(0.5601 0.6101 1u 1n 1n 1 1)"

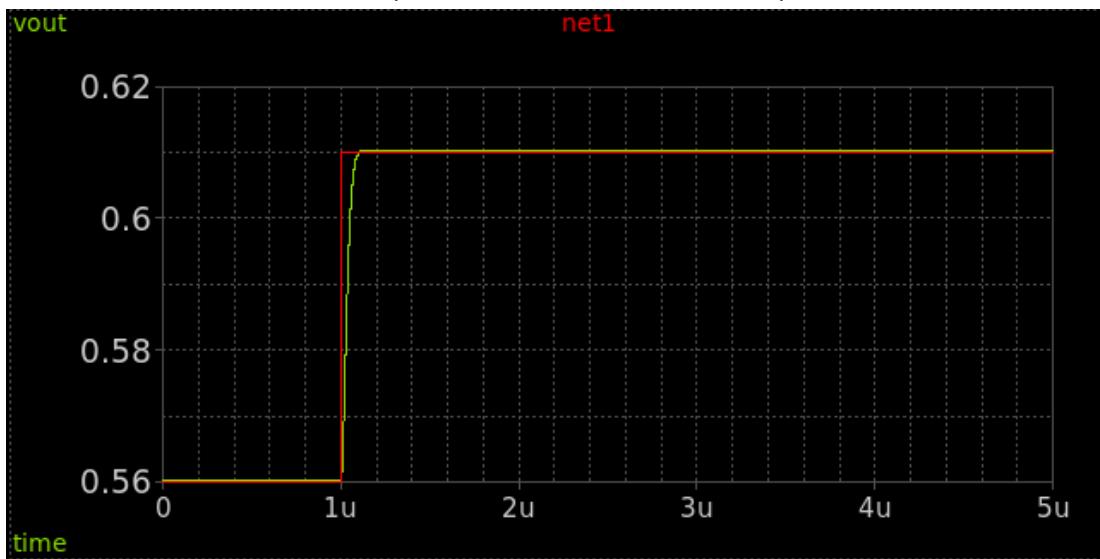


Figure 34 : Vin and Vout overlaid

```
No. of Data Rows : 50014
t_rise           = 5.053179e-08 targ= 1.059174e-06 trig= 1.008642e-06
t_rise = 5.053179e-08
```

Figure 35 : Time rise

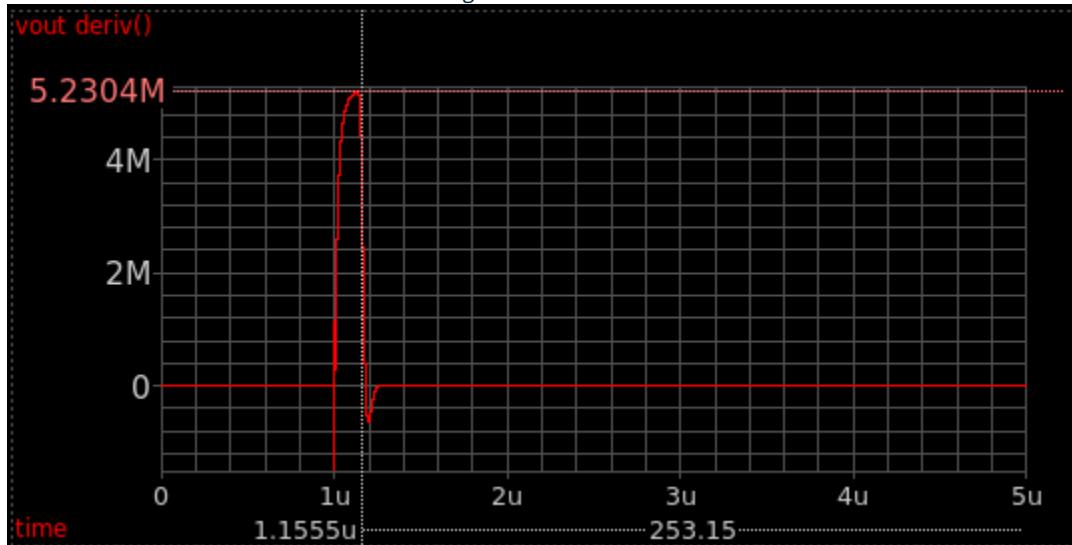


Figure 36 : drivetive of Vout

`slew_rate_vus = 5.029949e+00`

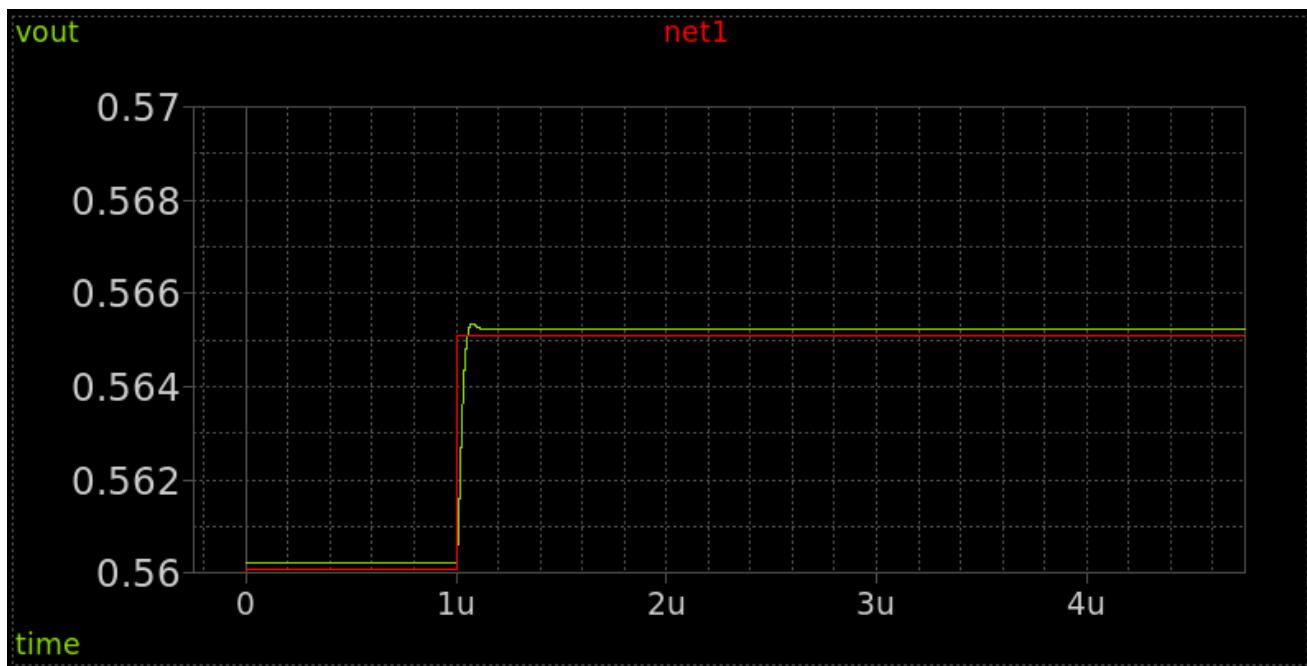
Figure 37 : Slew rate

Analytical :

$$SR = IB_1/CC = 9.7\mu/2p = 4.85 \text{ v/us}$$

Parameter	Hand Calculation (Analytical)	Simulation Result
SR	4.85 v/us	5.03 V/us

Settling time:

Figure 38 : *V_{in}* and *V_{out}* overlaid

```
t_rise          =  3.945044e-08 targ=  1.047183e-06 trig=  1.007732e-06
t_rise = 3.945044e-08
```

Figure 39 : Time rise

$$\text{rise time} = 2.2 * \text{taw} = 2.2 * 1 / w_u = 56.3 \text{ ns}$$

Parameter	Hand Calculation (Analytical)	Simulation Result
rise time	56.3 ns	39.45 ns

The simulation result will be better than expected. Why?

The result is better because the hand calculation uses a simple first-order model ($t_{rise}=2.2\tau$), while the simulated OTA is a faster-responding second-order system.

Do you see any ringing? Why?

Yes, there is a very slight overshoot because the phase margin of $\sim 70^\circ$ creates a slightly underdamped response

Part 5 (optional): DC Closed Loop AC Open-Loop OTA Simulation

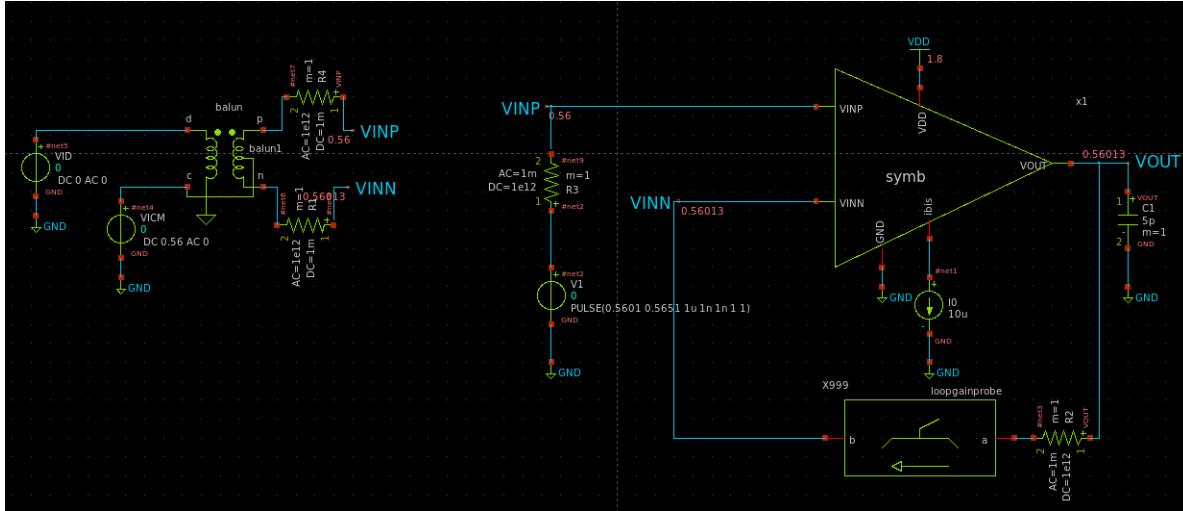


Figure 40 : Testbench of DC Closed Loop AC Open-Loop OTA Simulation

BSIM4v5: Berkeley Short Channel IGFET Model-4		
device	m.x1.xm8.m0	m.x1.xm7.m0
model	nmos_3p3.12	pmos_3p3.12
id	4.07546e-05	4.07546e-05
gm	0.000520237	0.000537349
gds	9.81766e-06	2.03236e-06
vgs	0.752069	0.880594
vth	0.681985	0.788012
vds	0.560125	1.23987
vdsat	0.125249	0.125964
BSIM4v5: Berkeley Short Channel IGFET Model-4		
device	m.x1.xm6.m0	m.x1.xm1.m0
model	pmos_3p3.12	pmos_3p3.12
id	9.6608e-06	4.82775e-06
gm	0.000127938	7.81288e-05
gds	9.44879e-07	5.66367e-07
vgs	0.880594	0.824168
vth	0.788761	0.781708
vds	0.415698	0.641115
vdsat	0.125462	0.0977928
BSIM4v5: Berkeley Short Channel IGFET Model-4		
device	m.x1.xm3.m0	m.x1.xm4.m0
model	nmos_3p3.8	nmox_3p3.8
id	4.82775e-06	4.83305e-06
gm	7.29312e-05	7.29929e-05
gds	5.97511e-07	5.95633e-07
vgs	0.743183	0.743183
vth	0.704146	0.704128
vds	0.743181	0.752069
vdsat	0.107147	0.107157

Figure 41 : OP Annotation

After achieving the correct DC operating point with all transistors in saturation, we ran the AC analysis and the results came out accurate and well-aligned with the design targets.

This table shows that your final Two-Stage Miller OTA design successfully meets or exceeds all the required performance targets

Specification	Requirement	Achieved Result	Status
DC Voltage Gain	≥ 66 dB	69.46 dB	✓
Gain-Bandwidth (GBW)	≈ 5 MHz	6.11 MHz	✓
CMRR @ DC	≥ 74 dB	80.46 dB	✓
Phase Margin	$\geq 70^\circ$	70.26°	✓
CM Input Range (Low)	≤ 0.2 V	0.176 V	✓
CM Input Range (High)	≥ 0.8 V	0.954 V	✓
Slew Rate	≥ 5 V/ μ s	5.03 V/ μ s	✓
Rise Time	≤ 70 ns	39.45 ns	✓

Appendix

OP code

```
.control
save all
op
show m : id : gm : gds : vgs : vth : vds : vdsat
print allv
write lab9_OP.raw
.endc
```

Diff small signal

```
.control
save all
op
ac dec 10 1 10G
meas ac Gain MAX vmag(VOUT) FROM=1 TO=10G
let ff2=Gain*0.707
meas ac BW WHEN vmag(VOUT)=ff2 FALL =1
save v(VOUT)
remzerovec
write lab9_ac.raw
.endc
```

Cm small signal

```
.control
save all
op
ac dec 10 1 10G
meas ac CMGain MAX vmag(VOUT) FROM=1 TO=1k
remzerovec
write lab9_ac.raw
.endc
```

CMRR

```
.control
save all
alter VID AC = 1
alter VICM AC = 0
ac dec 10 1 10g
alter VID AC = 0
alter VICM AC = 1
ac dec 10 1 10g
let vod = ac1.v(VOUT)
let vocm = ac2.v(VOUT)
let cmrr_freq = vdb(vod) - vdb(vocm)
write 5t_ota_tb_ac.raw
.endc
```

(Optional) Diff large signal ccs:

```
.control
save all
dc VID -0.1 0.1 0.0001
meas dc Vout_at0 find v(VOUT) when param=0
plot VOUT
plot deriv(VOUT)
save VOUT
remzerovec
write 5.raw
.endc
```

CM large signal ccs (region vs VICM):

```
.temp 27

*don't remove the '.temp 27' line but you can change the temp if you want
.control
*change the line below to the number of MOSFETs in your circuit
let number_of_transistors = 8

let x = 1
dowhile x <= number_of_transistors
  save @m.x1.xm{$&x}.m0[vgs]
  save @m.x1.xm{$&x}.m0[vds]
  save @m.x1.xm{$&x}.m0[vdsat]
  save @m.x1.xm{$&x}.m0[vth]
  let x = x + 1
end

DC VCM 0 1.8 10m
*you can run any type of analysis as long as it generates a vector

let x = 1
dowhile x <= number_of_transistors
  let index = 0
  let vgs = @m.x1.xm{$&x}.m0[vgs]
  let vth = @m.x1.xm{$&x}.m0[vth]
  let vds = @m.x1.xm{$&x}.m0[vds]
  let vdsat = @m.x1.xm{$&x}.m0[vdsat]
  let vtm_10 = 0.26*((temp+273)/300)
  let cut_off_con = vth - vtm_10
  let region_vec = vgs

  foreach val $&vgs
    if (vgs[index] <= cut_off_con)
      let regionp = 0
```

```

    else
      if (vgs[index] <= vth[index])
        let regionp = 3
      else
        if (vds[index] <= vdsat[index])
          let regionp = 1
        else
          let regionp = 2
      end
    end
  end
  let region_vec[index] = regionp
  let index = index + 1
end
let M{$&x}_op_region = region_vec
unlet vgs vth vds vdsat vtm_10    cutt_off_con @m.x1.xm{$&x}.m0[vgs]
+ @m.x1.xm{$&x}.m0[vth] @m.x1.xm{$&x}.m0[vds] @m.x1.xm{$&x}.m0[vdsat]
+ index regionp cutt_off_con
let x = x + 1
end
unlet region_vec
write regions.raw
.endc

```

(Optional) CM large signal ccs (GBW vs VICM) :

```

"
.control
save all
*0 cut-off
*1 triode
*2 sat
*3 subth
=====
*change this if you have a different number of mosfets (max = 12)
let transistor_count = 8
=====
let mos_loop = transistor_count
let vicm_start = -0.6
let vicm_stop = 1.8
let vicm_step = 20m
let num_points = (vicm_stop - vicm_start) / vicm_step
* Pre-allocate vectors
let vicm_vec = vector(num_points) *0
let gbw_vec = vector(num_points) *0
let region_m1 = vector(num_points) *0
let region_m2 = vector(num_points) *0
let region_m3 = vector(num_points) *0

```

```
let region_m4 = vector(num_points) *0
let region_m5 = vector(num_points) *0
let region_m6 = vector(num_points) *0
let region_m7 = vector(num_points) *0
let region_m8 = vector(num_points) *0
let region_m9 = vector(num_points) *0
let region_m10 = vector(num_points) *0
let region_m11 = vector(num_points) *0
let region_m12 = vector(num_points) *0

* Define vectors type
settype voltage vicm_vec
setscale vicm_vec
gbw_vec
let vicm_val = vicm_start
let i = 0
set mos ( xm1 xm2 xm3 xm4 xm5 xm6 xm7 xm8 xm9 xm10 xm11 xm12 )
while vicm_val le vicm_stop
    * Set CM voltage and enable differential input
    alter VICM DC = vicm_val
    let x = 1
    dowhile x <= $&mos_loop
        save @m.x1.$mos[$&x].m0[id]
        save @m.x1.$mos[$&x].m0[vgs]
        save @m.x1.$mos[$&x].m0[vds]
        save @m.x1.$mos[$&x].m0[vdsat]
        save @m.x1.$mos[$&x].m0[vth]
        let x = x + 1
    end

    op

    let y = 1

    dowhile y <= transistor_count
        set num = {$&y}
        let id = (@m.x1.xm{$num}.m0[id])
        let vgs = (@m.x1.xm{$num}.m0[vgs])
        let vds = (@m.x1.xm{$num}.m0[vds])
        let vth = (@m.x1.xm{$num}.m0[vth])
        let vdsat = (@m.x1.xm{$num}.m0[vdsat])
        let vtm = 0.026
        let subth_low = vth - 10*vtm
        let region = 0

        if ($&vgs >= $&vth & $&vds > $&vdsat )
            let region = 2
```

```

else
    if ($&vgs >= $&subth_low & $&vgs < $&vth)
        let region = 3
    else
        if ($&vgs >= $&vth & $&vds < $&vdsat)
            let region = 1
        else
            if ($&id < 6.3e-9 | $&vgs < 1e-10)
                let region = 0
            else
                let region = 1
            end
        end
    end
end

let region_m{$&y}[i] = region

let y = y + 1
end
* AC to get DC gain and BW
ac DEC 10 1 10G
meas AC gain MAX vmag(VOUT)
let stop_band=gain*0.707
meas AC bw WHEN vmag(VOUT)=stop_band
* Store results in pre-allocated vectors
let vicm_vec[i] = vicm_val
let gbw_vec[i] = gain * bw
let vicm_val = vicm_val + vicm_step
let i = i + 1
end

*measuring
=====
meas AC max_gbw max f(gbw_vec)
let limit_gbw = 0.9*max_gbw
meas AC max_vincm find v(vicm_vec) when f(gbw_vec)=limit_gbw FALL=1
meas AC min_vincm find v(vicm_vec) when f(gbw_vec)=limit_gbw RISE=1
let CMIR=max_vincm-min_vincm
print CMIR
save vicm_vec gbw_vec

*scaling the region vectors
=====
let region_m1 = region_m1 * max_gbw / 3
let region_m2 = region_m2 * max_gbw / 3
let region_m3 = region_m3 * max_gbw / 3

```

```

let region_m4 = region_m4 * max_gbw / 3
let region_m5 = region_m5 * max_gbw / 3
let region_m6 = region_m6 * max_gbw / 3
let region_m7 = region_m7 * max_gbw / 3
let region_m8 = region_m8 * max_gbw / 3
let region_m9 = region_m9 * max_gbw / 3
let region_m10= region_m10 * max_gbw / 3
let region_m11= region_m11 * max_gbw / 3
let region_m12= region_m12 * max_gbw / 3
write Lab9_CMIR.raw vicm_vec gbw_vec region_m1 region_m2 region_m3 region_m4 region_m5
region_m6 region_m7 region_m8
.endc

```

Loop gain

```

.func tian_loop() {1/(1-1/(2*(ac1.I(v.X999.Vi)*ac2.V(X999.x)-
ac1.V(X999.x)*ac2.I(v.X999.Vi))+ac1.V(X999.x)+ac2.I(v.X999.Vi)))}

*-----*
* Probes
*-----*
.save V(X999.x) I(v.X999.Vi)

*-----*
* CONTROL BLOCK
*-----*

.control
  set num_threads=8
  set color0=white
  set color1=black
  set xbrushwidth =3
  unset askquit

  optran 0 0 0 100n 4u 0
  op
  write stb.raw
  set appendwrite

*-----*
* LSTB analysis
*-----*
* Set voltage AC to 1
ac dec 50 100 10G

* Set Current to 1

```

```

alter i.X999.Ii acmag=1
alter v.X999.Vi acmag=0
ac dec 50 100 10G

let tian_signal = tian_loop()

let loop_gain_db = db(tian_signal)
let loop_gain_lin = mag(tian_signal)
let loop_gain_ph = 180*cph(tian_signal)/pi

*--- DC gain (low frequency gain, usually at first AC point ~100 Hz) ---
meas ac DC_GAIN_DB FIND loop_gain_db AT=100
meas ac DC_GAIN_LIN FIND loop_gain_lin AT=100

*--- Bandwidth (-3 dB frequency relative to DC gain) ---
let target_db = DC_GAIN_DB - 3
meas ac BW WHEN loop_gain_db=target_db

*--- Gain Bandwidth Product ---
let GBW = DC_GAIN_LIN * BW
print GBW

*--- Unity-gain frequency fu (still useful) ---
meas ac FU WHEN loop_gain_db=0

*--- Phase margin at unity gain ---
meas ac PM FIND loop_gain_ph AT=FU
print PM

Remzerovec
write stb.raw tian_signal

.endc

```

Slew rate & Settling time:

```

".control
save all

```

```
tran 0.1n 5u
```

```
let v_initial = x.  
let v_final = x + margin  
let v_swing = v_final - v_initial  
  
let v_10_percent = v_initial + 0.1 * v_swing  
let v_90_percent = v_initial + 0.9 * v_swing  
  
meas tran t_rise TRIG v(VOUT) VAL=v_10_percent RISE=1 TARG v(VOUT) VAL=v_90_percent RISE=1  
  
print t_rise  
  
write rise_time.raw  
.endc"
```