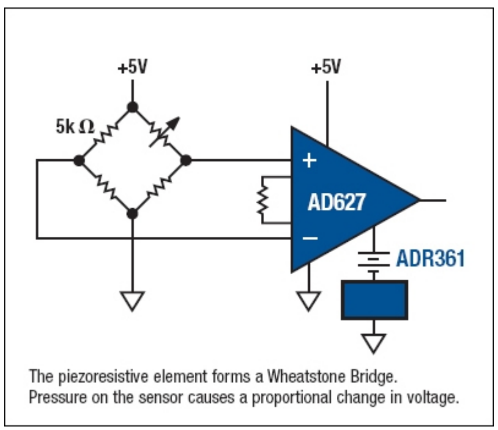
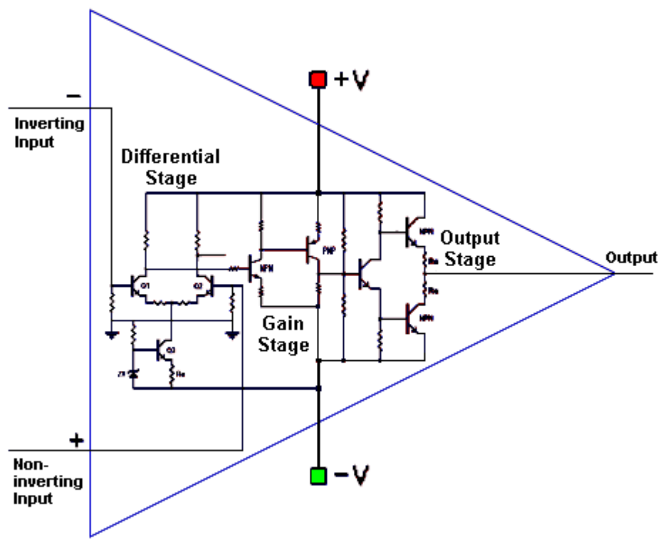
Differential Pair

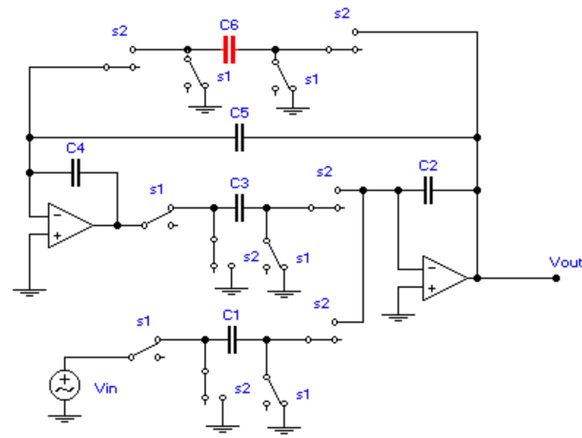
Claudio Talarico, Gonzaga University
Fall 2014

Significance of the Differential Pair

- The differential pair is the most widely used two-transistor circuit in integrated circuits
  - Many circuits rely on the “computation” of the difference between two voltages

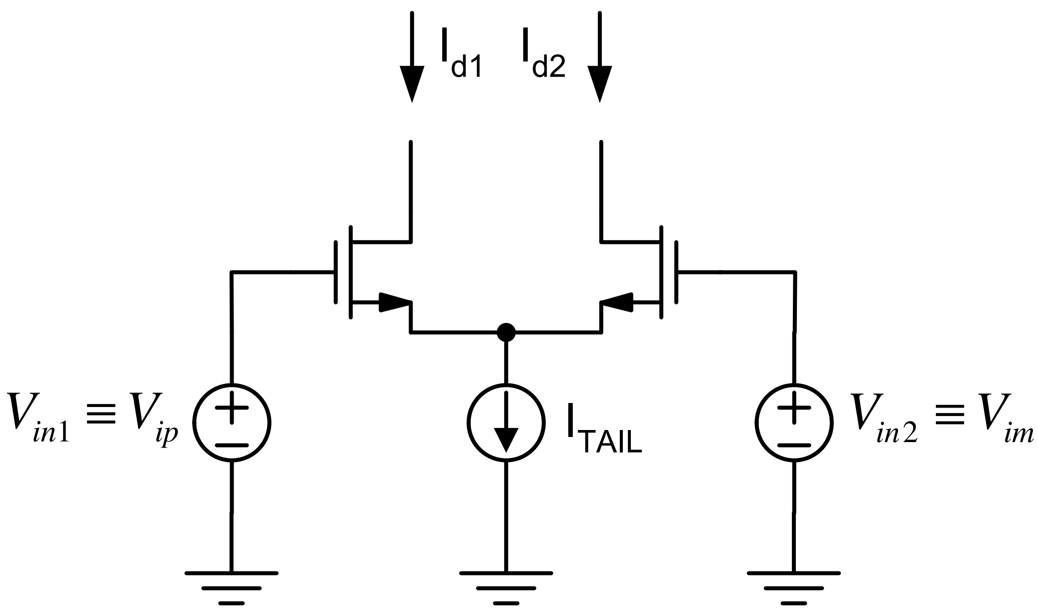






source: R. Dutton, B. Murmann

General Analysis of the Differential Pair



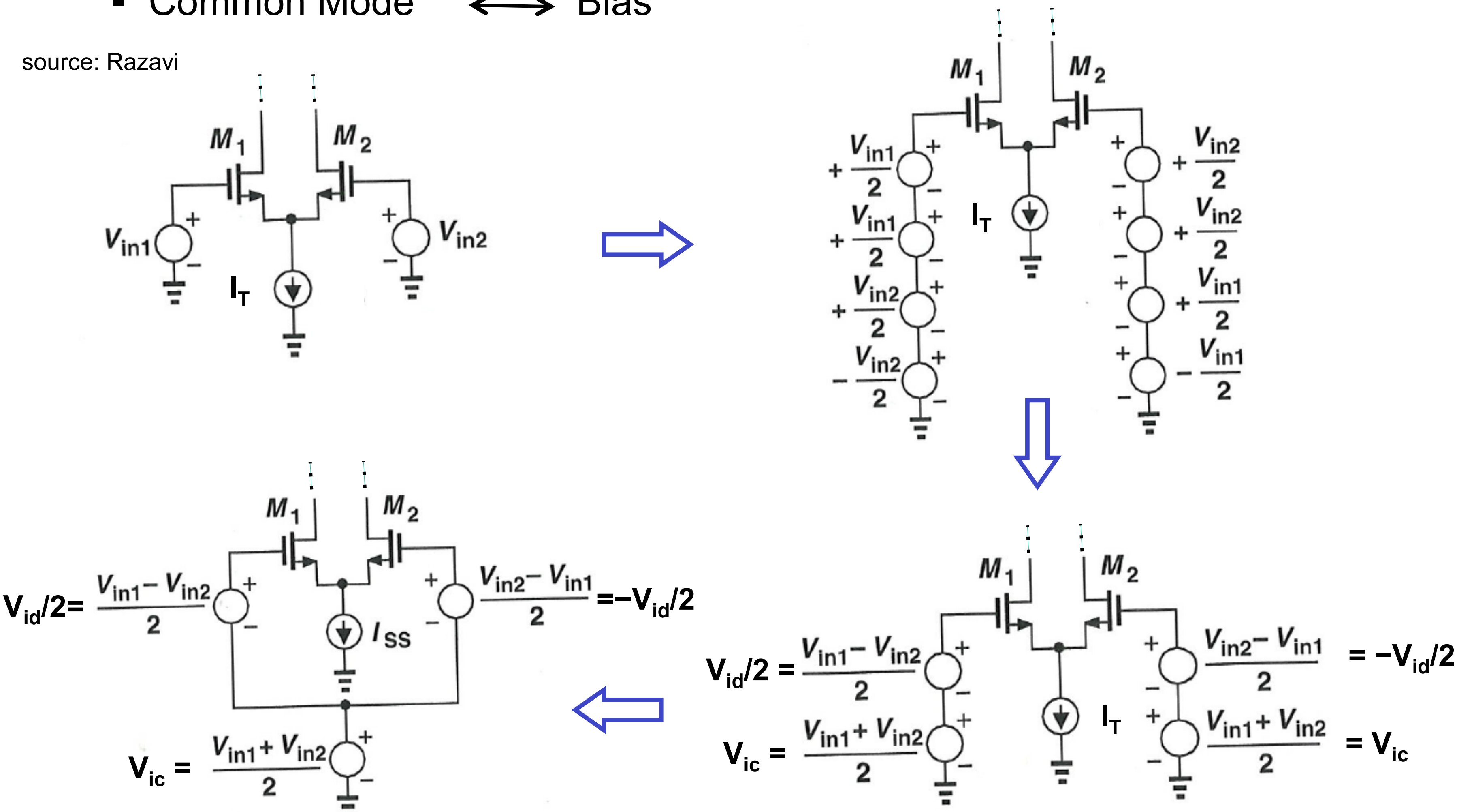
- When $V_{ip}= V_{im}$, and both transistors are identical, we must have $I_{d1}=I_{d2}=I_{TAIL}/2$
- How about $V_{ip}=V_{im}=1V$ versus $V_{ip}=V_{im}=2V$?
  - Makes no difference !
- From a signal perspective, we care only about the difference of the applied voltages
  - Makes sense to introduce a new variable:
    - $V_{id} = V_{ip} - V_{im}$

source: R. Dutton, B. Murmann

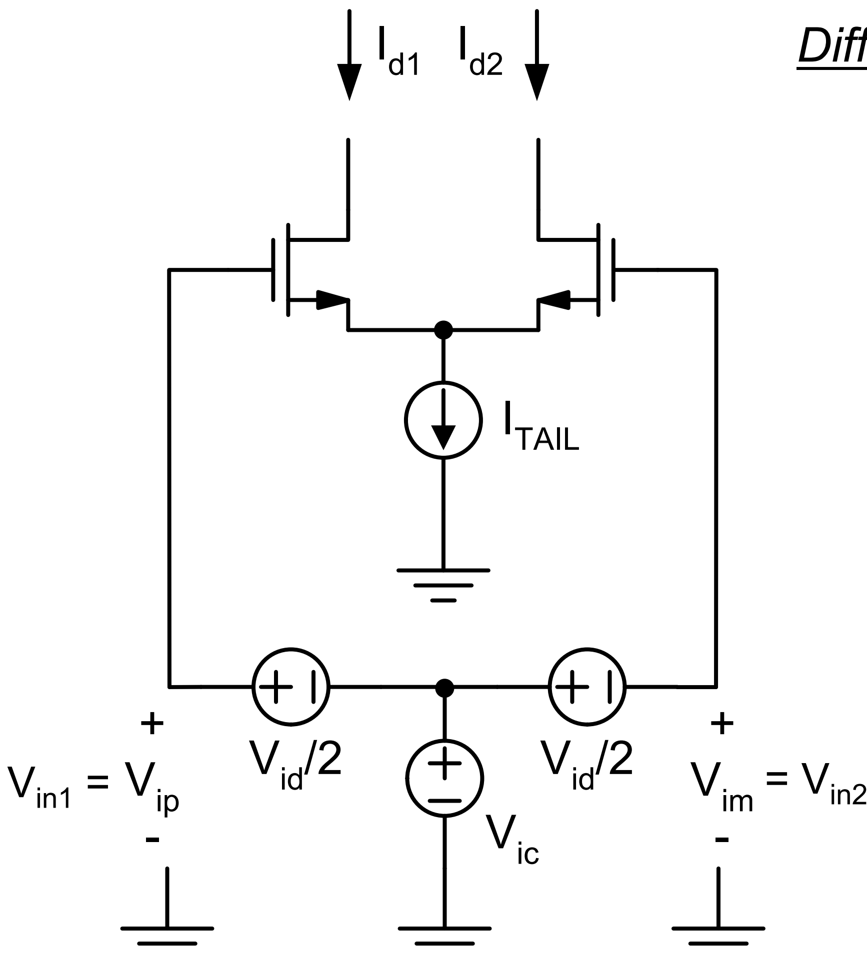
Differential Mode and Common Mode (1)

- Differential Mode ⟷ Signal
- Common Mode ⟷ Bias

source: Razavi



Differential Mode and Common Mode (2)



Difference

$$V_{id} = V_{ip} - V_{im}$$

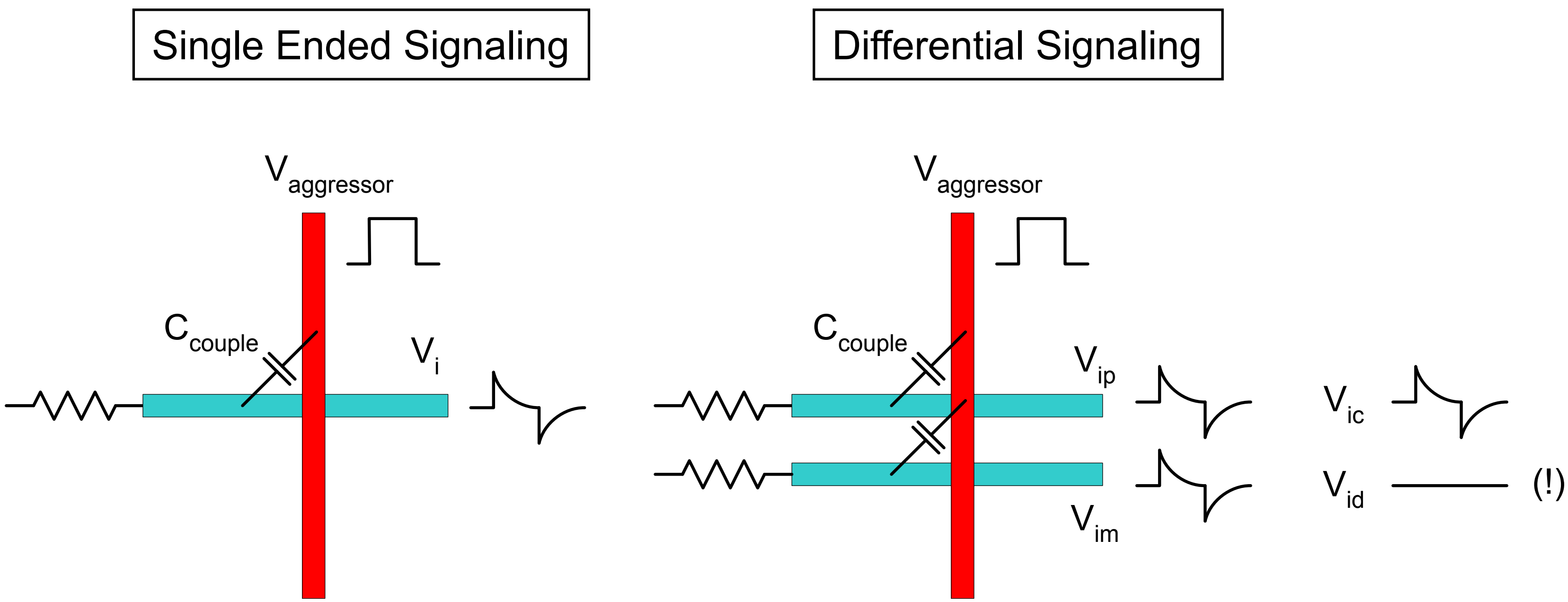
$$V_{ip} = V_{ic} + \frac{V_{id}}{2}$$

$$V_{im} = V_{ic} - \frac{V_{id}}{2}$$

Average

$$V_{ic} = \frac{V_{ip} + V_{im}}{2}$$

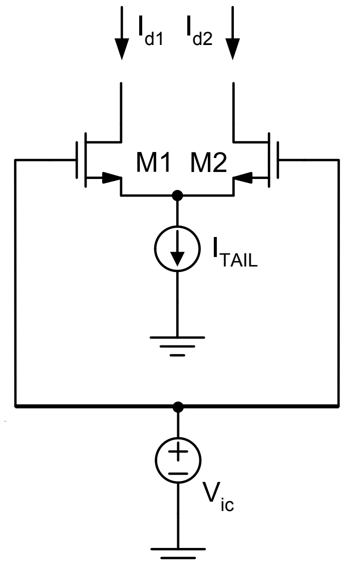
Differential Signaling and Coupling Noise Immunity



Similar arguments can be made regarding the rejection of supply variation, ground bounce, etc.

Differential Pair Bias (Common Mode Bias)

- We are interested in the range of input DC bias (V_{IC}) over which the differential pair operates properly



- The highest value of V_{IC} (= V_{ICmax}) is limited by the requirement that M_1 and M_2 remain in saturation

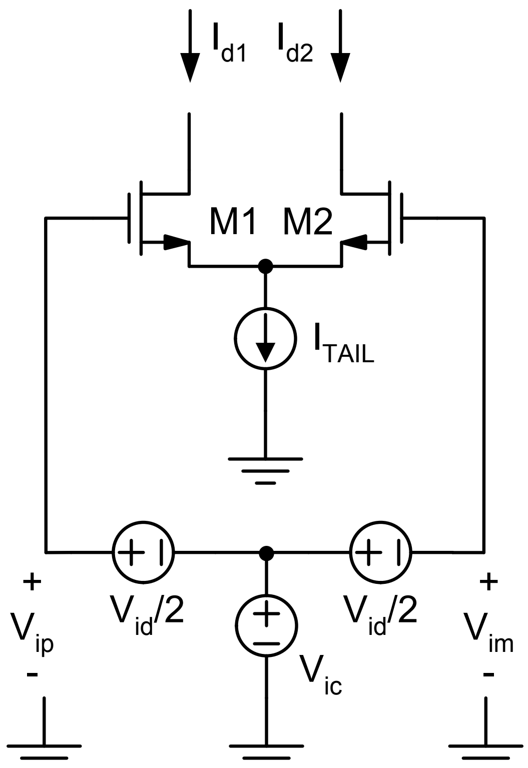
$$\cancel{V_{GS}} - V_t < V_{DS} = V_D - V_{IC} + \cancel{V_{GS}} \longleftrightarrow V_{IC} < V_t + V_D$$

$$V_{IC} < V_t + V_{DD} - \frac{I_{TAIL}}{2} R$$

- The lowest value of V_{IC} (= V_{ICmin}) is determined by the need to have sufficient voltage across the tail current source for it to operate properly (M_{TAIL} in saturation)

$$V_{IC} > V_t + V_{OV} + V_{OVTAIL}$$

Large Signal Transfer Function (1)



$$V_{ip} - V_{gs1} = V_{im} - V_{gs2} \Leftrightarrow V_{id} \equiv V_{ip} - V_{im} = V_{gs1} - V_{gs2}$$

$$I_{d1} + I_{d2} = I_{TAIL}$$

$$V_{gs1} = V_t + \underbrace{\sqrt{\frac{2I_{d1}}{\mu C_{ox}\frac{W}{L}}}}_{= V_{OV1}} \qquad V_{gs2} = V_t + \underbrace{\sqrt{\frac{2I_{d2}}{\mu C_{ox}\frac{W}{L}}}}_{= V_{OV2}}$$

$$V_{id} = \frac{\sqrt{I_{d1}} - \sqrt{I_{d2}}}{\sqrt{\frac{\mu C_{ox}}{2}\frac{W}{L}}}$$

* see aside for the "algebra"

$$I_{od} = I_{d1} - I_{d2} = \frac{1}{2}\mu C_{ox}\frac{W}{L}V_{id}\sqrt{\frac{4I_{TAIL}}{\mu C_{ox}\frac{W}{L}} - V_{id}^{\,2}}$$

source: R. Dutton, B. Murmann

Aside *

Square both terms of the following equation and solve the resulting second order equation (for the unknown $\sqrt{I_{d1}}$)

$$V_{id} = \frac{\sqrt{I_{d1}} - \sqrt{I_{d2}}}{\sqrt{\frac{\mu C_{ox}}{2}\frac{W}{L}}} \quad \Rightarrow \quad \begin{cases} I_{d1} = \frac{I_{TAIL}}{2} + \frac{\mu C_{ox}}{4}\frac{W}{L}V_{id}\sqrt{\frac{4I_{TAIL}}{\mu C_{ox}\frac{W}{L}} - V_{id}^2} \\ I_{d2} = \frac{I_{TAIL}}{2} - \frac{\mu C_{ox}}{4}\frac{W}{L}V_{id}\sqrt{\frac{4I_{TAIL}}{\mu C_{ox}\frac{W}{L}} - V_{id}^2} \end{cases}$$

$I_{d2} = I_{TAIL} - I_{d1}$

for $V_{id} > 0$ ⟶ I_{d1} must be > $I_{TAIL}/2$ (and I_{d2} must be < than $I_{TAIL}/2$ by the same amount)

for $V_{id} = 0$ ⟶ $I_{d1} = I_{d2} = I_{TAIL}/2$

Large Signal Transfer Function (2)

- We can turn this into a more elegant expression by using

$$\frac{I_{TAIL}}{2} = \frac{1}{2}\mu C_{ox}\frac{W}{L}V_{OV}{}^{2}$$

where V_{OV} is the quiescent point gate overdrive with V_{id}=0

$$\boxed{\frac{I_{od}}{I_{TAIL}} = \frac{I_{d1}-I_{d2}}{I_{TAIL}} = \frac{V_{id}}{V_{OV}}\sqrt{1-\left(\frac{V_{id}}{2V_{OV}}\right)^{2}}}$$

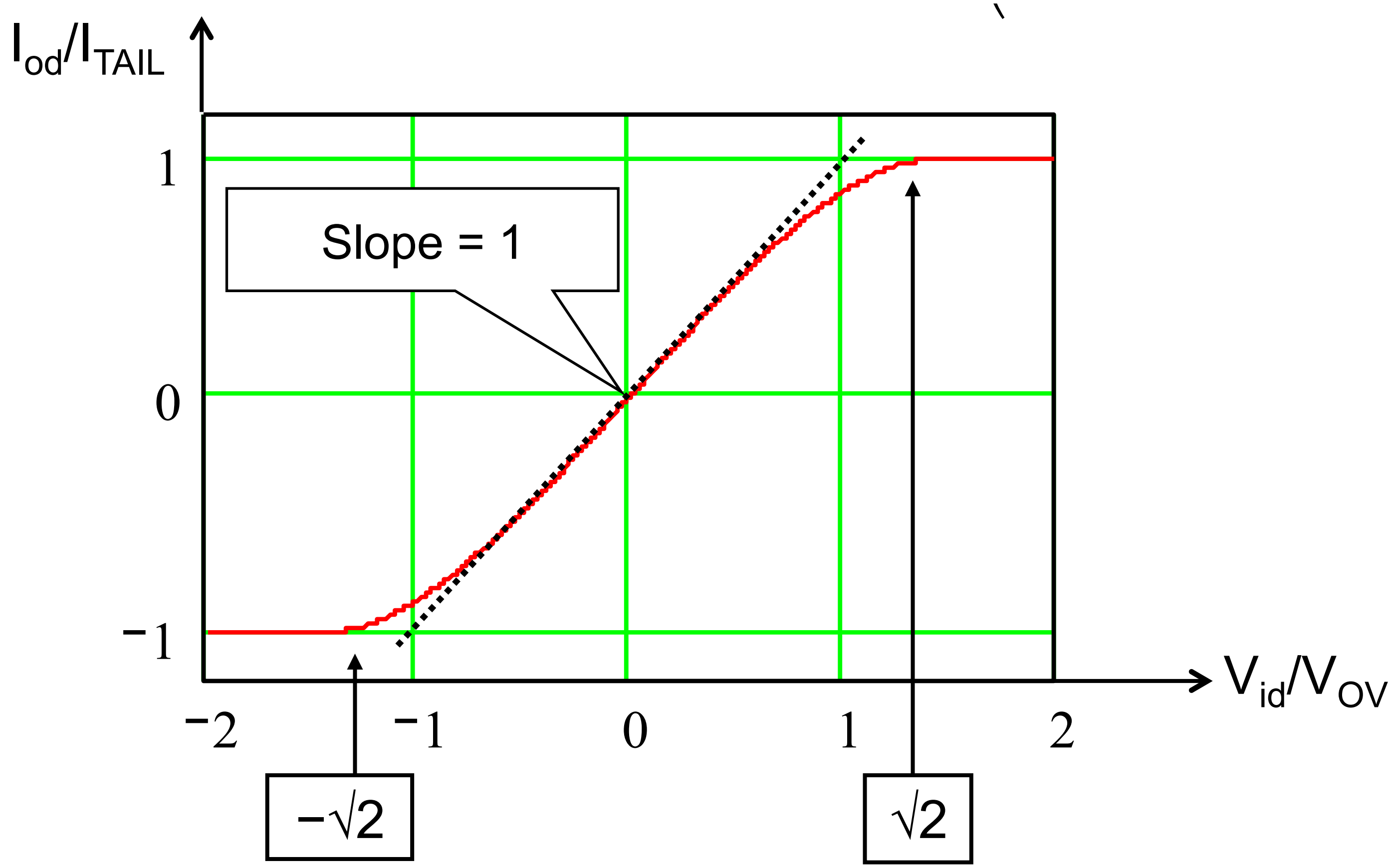
With V_{id}=0:
I_{d1}=I_{d2}=I_{TAIL}/2

$$V_{OV} \equiv \sqrt{\frac{I_{TAIL}}{\mu C_{ox}\frac{W}{L}}}$$

- This equation predicts
  - I_{od}/I_{TAIL} = 0 when V_{id}=0, as expected
  - Complete current steering (I_{od}/I_{TAIL}=±1) takes place when V_{id}= $\pm V_{OV}\sqrt{2}$

Large Signal Plot

- Note that the equation on previous slide is only valid in the center of this transfer function (that is for $|V_{id}| \leq \sqrt{2}\, V_{OV}$))
  - The equation assume both M_1 and M_2 are in saturation



source: R. Dutton, B. Murmann

Observations

- The large Signal TF Looks like something we have seen before
  - A transfer function that is somewhat linear as long as $V_{id} << 2\,V_{OV}$

$$\boxed{\frac{I_{od}}{I_{TAIL}} = \frac{I_{d1} - I_{d2}}{I_{TAIL}} = \frac{V_{id}}{V_{OV}}\sqrt{1-\left(\frac{V_{id}}{2V_{OV}}\right)^2}} \simeq \frac{V_{id}}{V_{OV}}$$

- For small signal analysis, we can find an equivalent transconductance by differentiation at the operating point

$$G_m = \left.\frac{dI_{od}}{dV_{id}}\right|_{V_{id}=0} = \frac{I_{TAIL}}{V_{OV}}$$

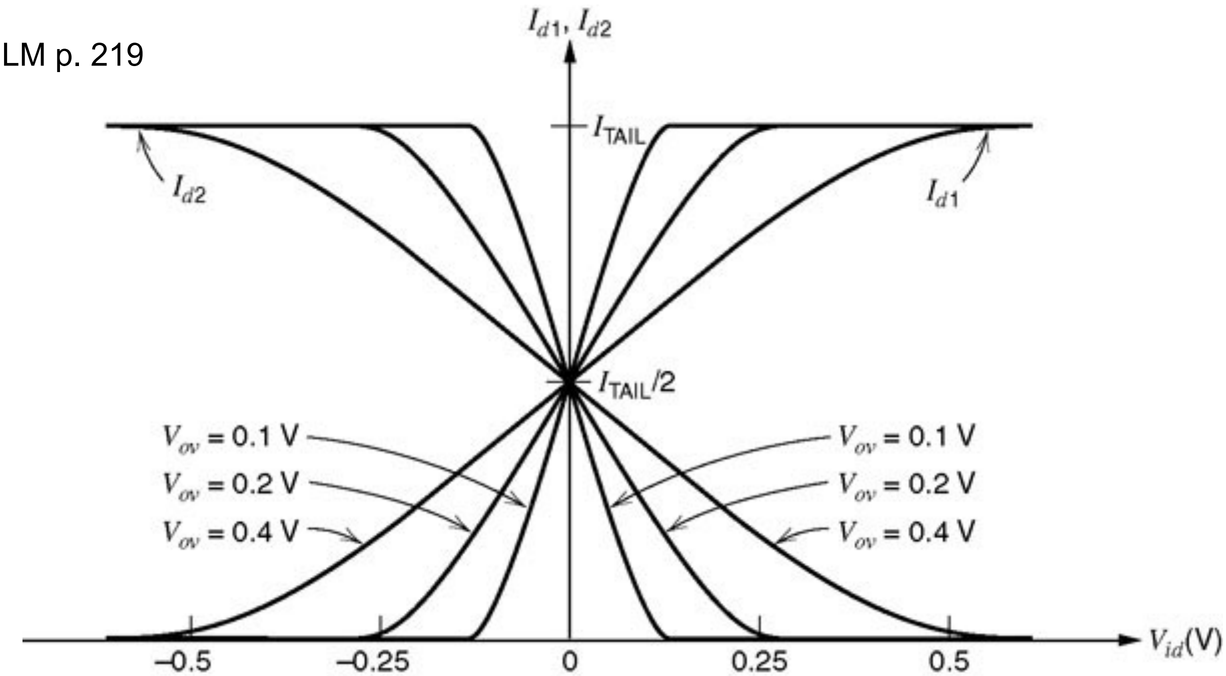
- Note that the transconductance of M_1 and M_2 is given by

$$g_{m1,2} = \frac{2I_D}{V_{OV}} = \frac{2\,\frac{I_{TAIL}}{2}}{V_{OV}} = \frac{I_{TAIL}}{V_{OV}}$$

Linearity range of the differential pair

- To increase the linearity range ($V_{id} << 2\ V_{OV}$) of a differential pair we need to increase V_{OV}
  - V_{OV} can be adjusted by changing W/L ratio or I_{TAIL}
    - For a fixed I_{TAIL} to increase V_{OV} we have to reduce the W/L ratio → the price is a reduction in g_m (g_m = slope of I_{od} vs. V_{id} T.F. plot). A reduction in g_m causes a reduction in gain
    - For obtaining a higher g_m the bias current I_{TAIL} can be increased → the cost is an increase in power consumption

source: GHLM p. 219



"Linear" Region of Transfer Function

- The requirement for "linear" T.F. can be generalized as follows to capture both long and short channel cases simultaneously:

$$V_{id} << \frac{2}{\left(\frac{g_m}{I_{TAIL}}\right)}$$

- A few approximate rule of thumbs:
  - For ~ 1.5% non linearity error

$$V_{id} < 0.5\frac{2}{\left(\frac{g_m}{I_{TAIL}}\right)}$$

  - For ~ 0.1% non linearity error

$$V_{id} < 0.2\frac{2}{\left(\frac{g_m}{I_{TAIL}}\right)}$$

source: B. Murmann

Another convenient property of the differential pair

- If M_1 and M_2 are identical and identical resistors are connected to the drains of M_1 and M_2 when V_{id}=0 then $I_{d1}-I_{d2} = 0$. This shows that when V_{id}=0 (no signal applied) than V_{od}=0 ($V_{od1}=V_{od2}=V_{DD}-R*I_{TAIL}/2$)

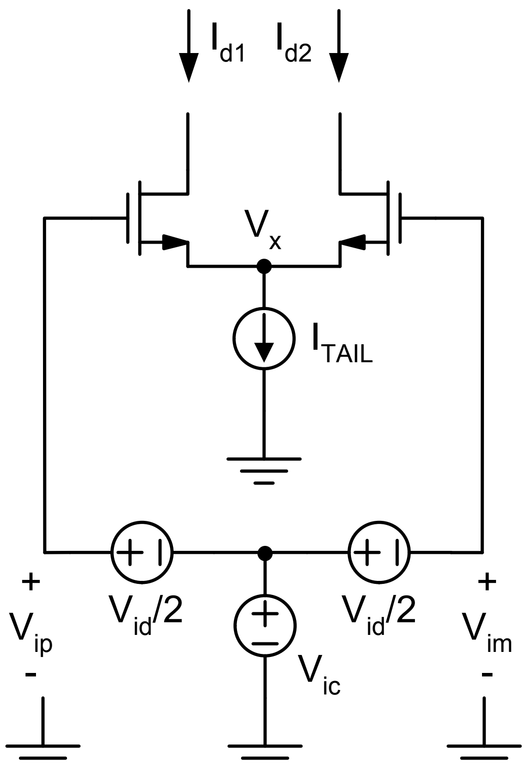
- This property allows direct coupling (no coupling capacitors) and biasing of cascaded MOS differential pairs

Does the Tail Node X Move?

$$KVL@V_X \Rightarrow V_X = V_{ic} + \frac{V_{id}}{2} - V_{gs1} = V_{ic} + \frac{V_{id}}{2} - V_t - \sqrt{\frac{2I_{d1}}{\mu C_{ox}\frac{W}{L}}}$$

Can show that:

$$V_x = V_{ic} - V_t - V_{OV}\sqrt{1 - \left(\frac{V_{id}}{2V_{OV}}\right)^2}$$



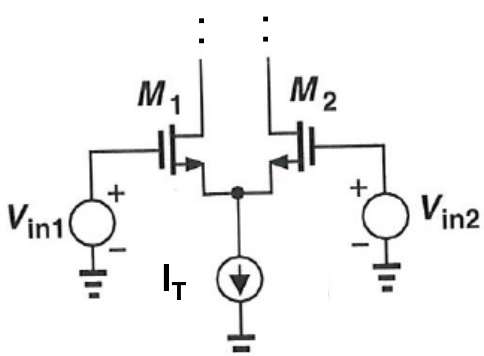
- From this expression, we see that from a small signal perspective the tail node is pinned at $V_{ic}-V_t-V_{OV}$
  - "AC ground"

source: R. Dutton, B. Murmann

Aside: A detailed look at the large signal DM TF (1)

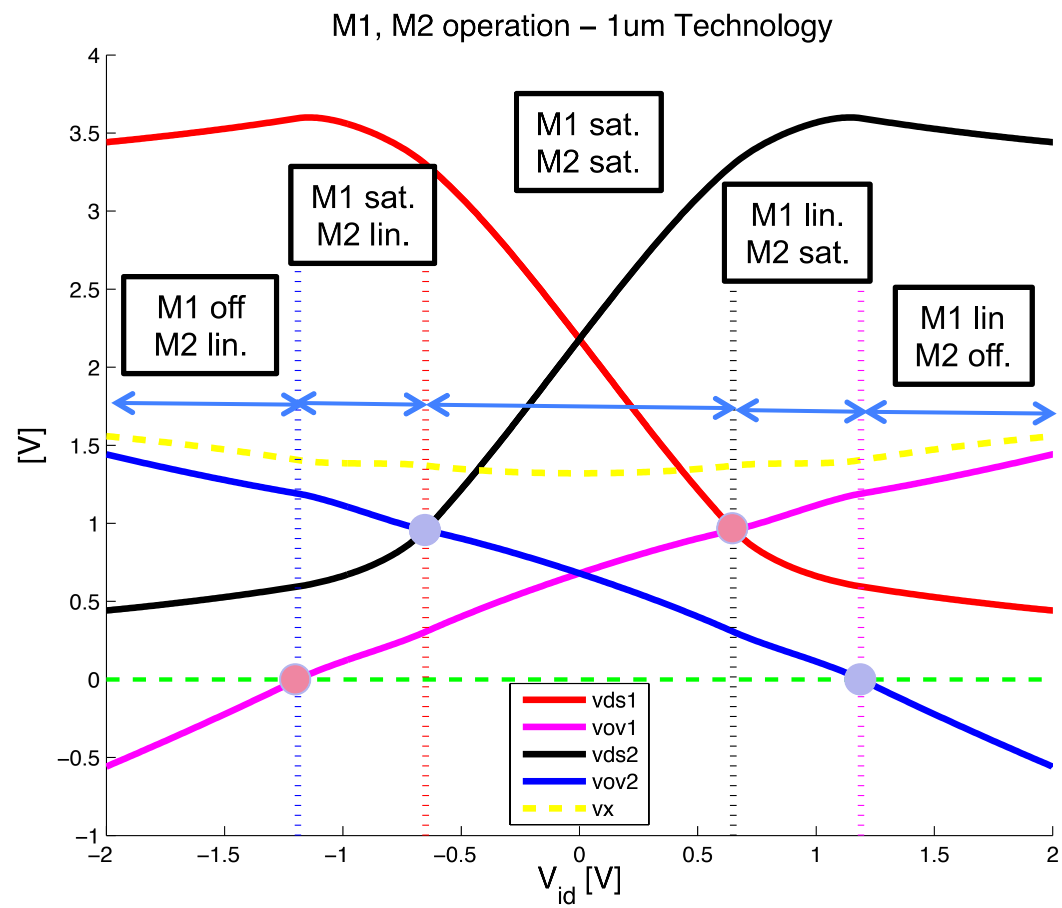
Example:

V_{DD} = 5V; I_T=600μA;
W/L=21.33μm/1μm;
R_L=5KΩ; (V_{IC}=2.5V)

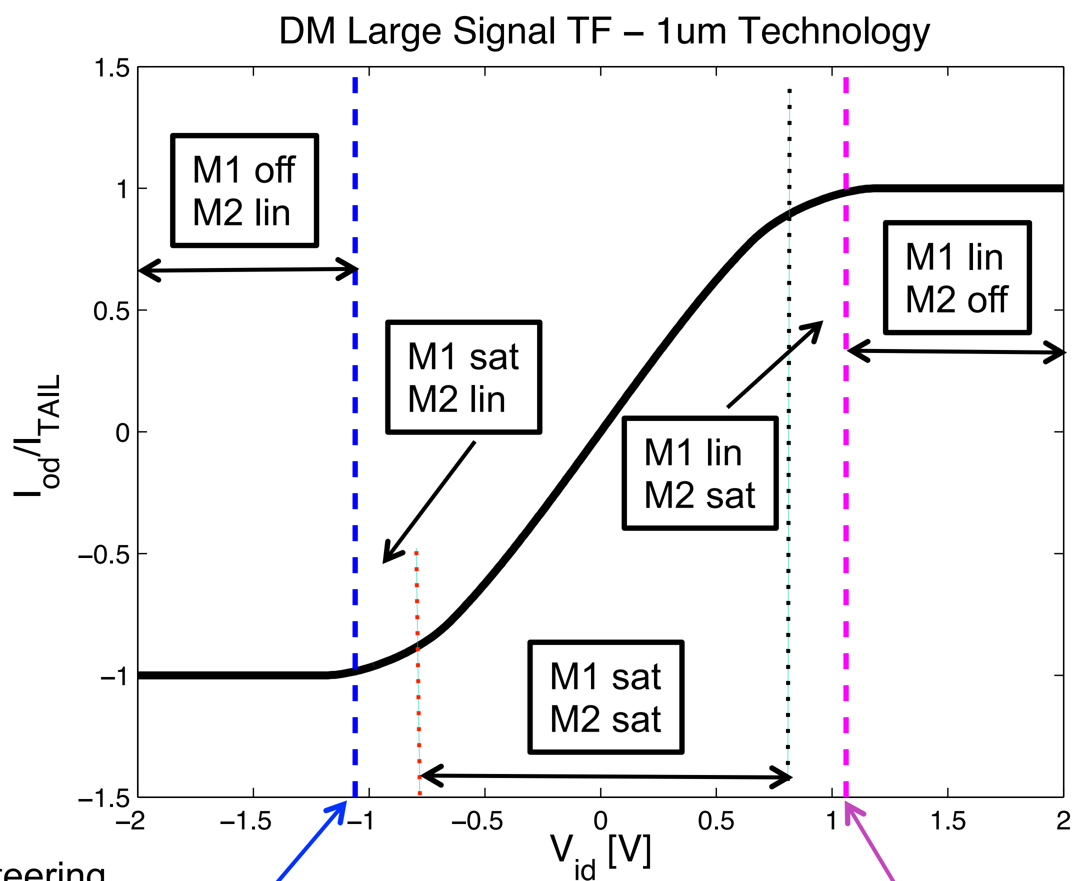


- $V_{id} = V_{in1} - V_{in2}$
- $V_{ic} = (V_{in1} + V_{in2})/2$
- $V_{od} = V_{o1} - V_{o2}$

Vid	M1	M2
Vid < −1.19	off	lin.
−1.19 < Vid < −0.65	sat	lin.
−0.65 < Vid < 0.65	sat.	sat.
0.65 < Vid < 1.19	lin.	sat
Vid > 1.19	lin.	off



Aside: A detailed look at the large signal DM TF (2)



≈ $-V_{OV,eq}\sqrt{2}$ (= −1.1V)

≈ $+V_{OV,eq}\sqrt{2}$ (= +1.1V)

complete current steering (I_{od}/I_{TAIL} = ±1) takes place when $V_{id} \approx \pm V_{OV,eq}\sqrt{2}$

<u>*approximate assumption*</u>: one MOST is in sat., and the other one is at the edge of cut off

$$V_{OV,eq} = \left(V_{GS} - V_{TH}\right)\Big|_{V_{id}=0} = \sqrt{\frac{I_{TAIL}}{\mu C_{OX}\frac{W}{L}}}$$

(= 750mV)

Aside: spice netlist

```
*
* ideal_dp.sp
* no backgate effect and ideal tail
*

* include models
.include ~/hspice_libs/ee114_hspice.mod

.option post brief accurate

* ideal balun
* example instantiation
* x1 vdm vcm vp vm balun
.subckt balun vdm vcm vp vm
e1 vp vcm transformer vdm 0 2
e2 vcm vm transformer vdm 0 2
.ends balun

* circuit
vdd vdd 0 5
vic vic 0 DC 2.5
vid vid 0 AC 1 DC=0

x1 vid vic vip vim balun
x2 vod voc vop vom balun
rdum vod 0 1gig * cannot leave the node floating

it vx 0 600u

m1 vop vgp vx vx nmos114 W=21.33u L=1u
m2 vom vgm vx vx nmos114 W=21.33u L=1u
RLp vdd vop 5K
RLm vdd vom 5K
CLp vop 0 50f
CLm vom 0 50f
RSp vip vgp 10k
RSm vim vgm 10k
```

```
* analysis and measurements
.OP
.ac dec 100 1e6 1e12
.pz v(vod) vid
.dc vid -5 5 0.01

.measure AC GDC find VDB(vod) at 1Meg
.measure AC ADC find V(vod) at 1Meg
.measure AC F3DB when VDB(vod)='GDC - 3'

.probe i(m1) i(m2) vth(m1) vth(m2) vdsat(m1) vdsat(m2)
.probe vgs(m1) vgs(m2) vds(m1) vds(m2)

* double check transistors' regions of operation
.alter
vid vid 0 AC=1 DC=-2    * M1 off, M2 linear
.alter
vid vid 0 AC=1 DC=-900m * M1 sat, M2 linear

.end
```

Aside: script to plot the simulation results (1)

```
%
% file: plot_ideal_dp.m
%
clear all; close all;
format short eng

addpath('/usr/local/MATLAB/HspiceToolbox');

% technology
RL = 5e3; IT=600e-6;vdd = 5; vt0 = 0.5;
W = 21.33e-6; L=1e-6; KP=50e-6;

figure(1)
y = loadsig('./spiceout/ideal_dp.sw0');
lssig(y)
vid = evalsig(y,'v_vid');
vod = evalsig(y,'v_vod');
vds1 = evalsig(y,'vds_m1');
vds2 = evalsig(y,'vds_m2');
vth1 = evalsig(y,'vth_m1');
vth2 = evalsig(y,'vth_m2');
vgs1 = evalsig(y,'vgs_m1');
vgs2 = evalsig(y,'vgs_m2');
vov1 = vgs1 - vth1;
vov2 = vgs2 - vth2;
vx = evalsig(y,'v_vx');
i1 = evalsig(y,'i_m1');
i2 = evalsig(y,'i_m2');
iod = i1-i2;
iod_it = iod/IT;
VOV_eq = sqrt(L*IT/KP/W)
plot(vid, iod_it,'linewidth',3, 'color', 'k', 'linestyle', '-');
xlim([-2 2]);
pointA = -sqrt(2)*VOV_eq
pointB = sqrt(2)*VOV_eq
% vertical lines
line([pointA pointA], [-1.5 1.5], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'b');
line([pointB pointB], [-1.5 1.5], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'm');
xlabel('V_{id} [V]', 'Fontsize', 14);
ylabel('I_{od}/I_{TAIL} ', 'Fontsize', 14);
title('DM Large Signal TF - 1um Technology', 'fontsize', 14);
```

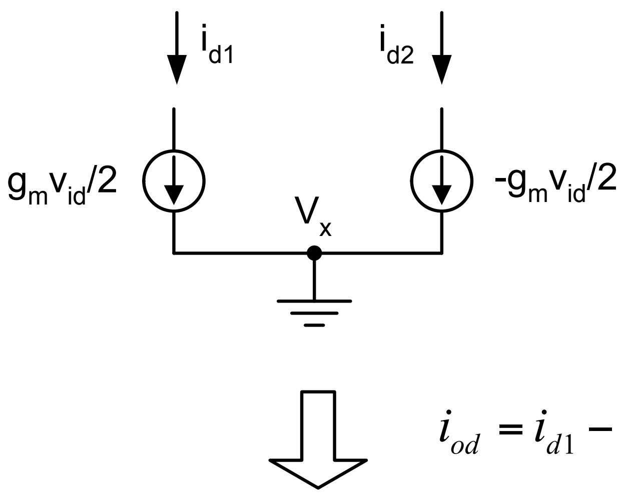
```
figure(2)
hold on;
xlim([-2 2]);
plot(vid, vds1,'linewidth',3, 'color', 'r', 'linestyle', '-');
plot(vid, vov1,'linewidth',3, 'color', 'm', 'linestyle', '-');
plot(vid, vds2,'linewidth',3, 'color', 'k', 'linestyle', '-');
plot(vid, vov2,'linewidth',3, 'color', 'b', 'linestyle', '-');
plot(vid, vx,'linewidth',3, 'color', 'y', 'linestyle', '--');
% horizontal line at 0
line([-2 2], [0 0], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'g');
legend('vds1','vov1','vds2','vov2', 'vx', 'location', 'south');
xlabel('V_{id} [V]', 'Fontsize', 14);
ylabel('[V]', 'Fontsize', 14);
title('M1, M2 operation - 1um Technology', 'fontsize', 14);
```

to be continued …

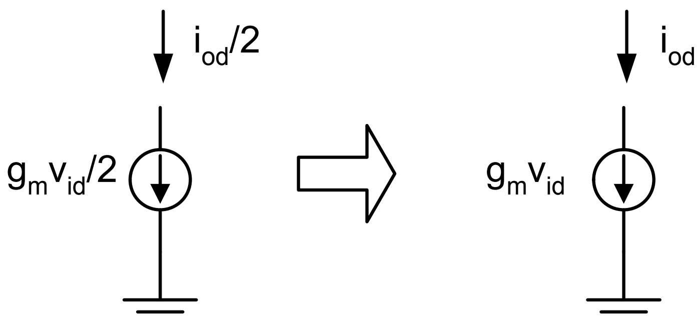
Aside: script to plot the simulation results (2)

```
% compute operating regions for M1
for i=1:length(vid)
  if vov1(i) > 0
     index = i;
     break
  end
end
% M1 exit cut off and enters saturation
M1off2sat = vid(index-1)
for i=1:length(vid)
  if vds1(i) < vov1(i)
     index = i;
     break
  end
end
% M1 exit saturation and enters triode
M1sat2triode = vid(index-1)
% vertical lines for M1
line([M1off2sat M1off2sat], [-1 3], 'linestyle', ':', ...
    'linewidth', 3, 'color', 'b');
line([M1sat2triode M1sat2triode], [-1 3], 'linestyle', ':', ...
    'linewidth', 3, 'color', 'k');

% compute operating regions for M2
for i=1:length(vid)
  if vds2(i) > vov2(i)
     index = i;
     break
  end
end
% M2 exit triode and enters saturation
M2triode2sat = vid(index)
for i=1:length(vid)
  if vov2(i) < 0
     index = i;
     break
  end
end
% M2 exit saturation and enters triode
M2sat2off = vid(index)
% vertical lines for M2
line([M2triode2sat M2triode2sat], [-1 3], 'linestyle', ':', ...
    'linewidth', 3, 'color', 'r');
line([M2sat2off M2sat2off], [-1 3], 'linestyle', ':', ...
    'linewidth', 3, 'color', 'm');
```

Small Signal Equivalent

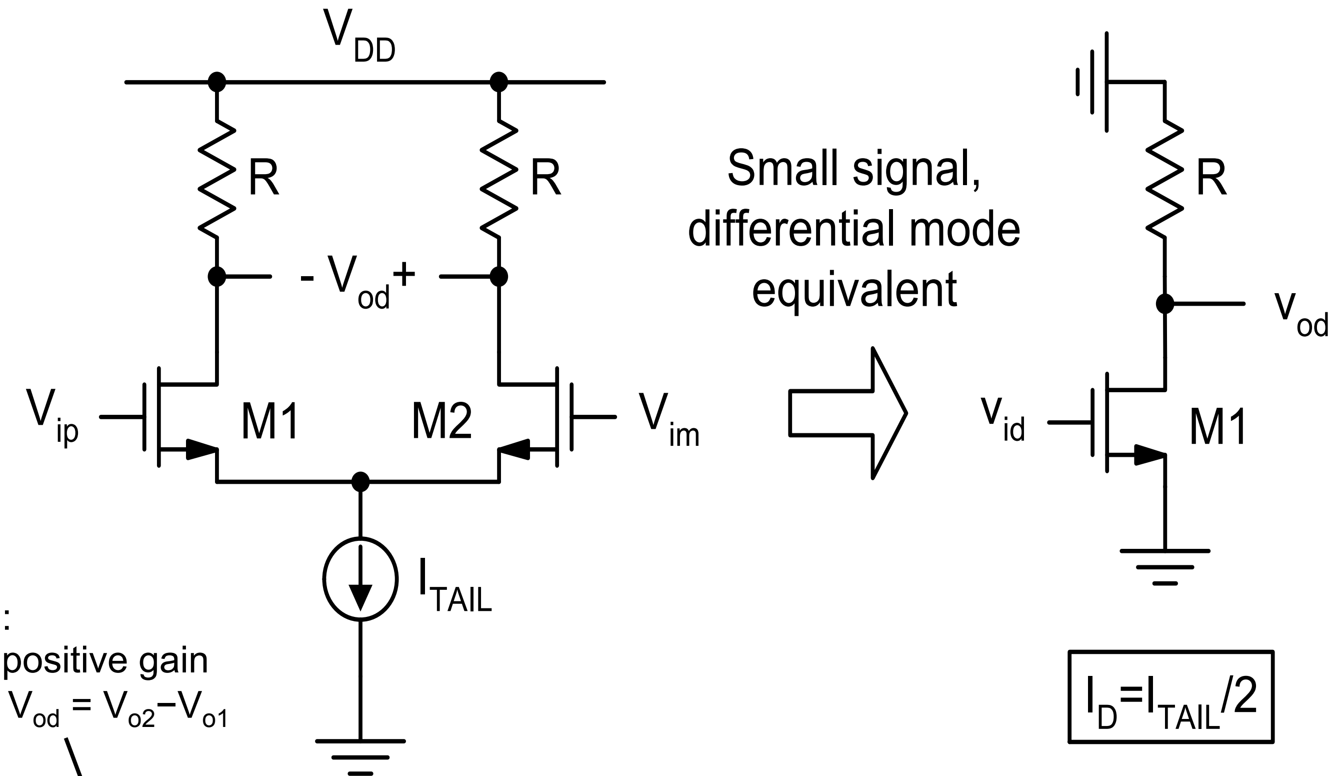


$$i_{od} = i_{d1} - i_{d2} = g_m v_{id}$$



- It is sufficient to work with half circuit!
  - Can directly apply everything we've learned about single transistor stage
- Half circuit caveats
  - Cannot analyze nonlinearity using half circuits
  - Assumes that M_1 and M_2 are identical

Basic Differential Voltage Amplifier Example



NOTE:
to get positive gain
chose $V_{od} = V_{o2} - V_{o1}$

$$A_{dm} = \frac{v_{od}}{v_{id}} = g_m (R \,||\, r_o) \approx g_m R$$

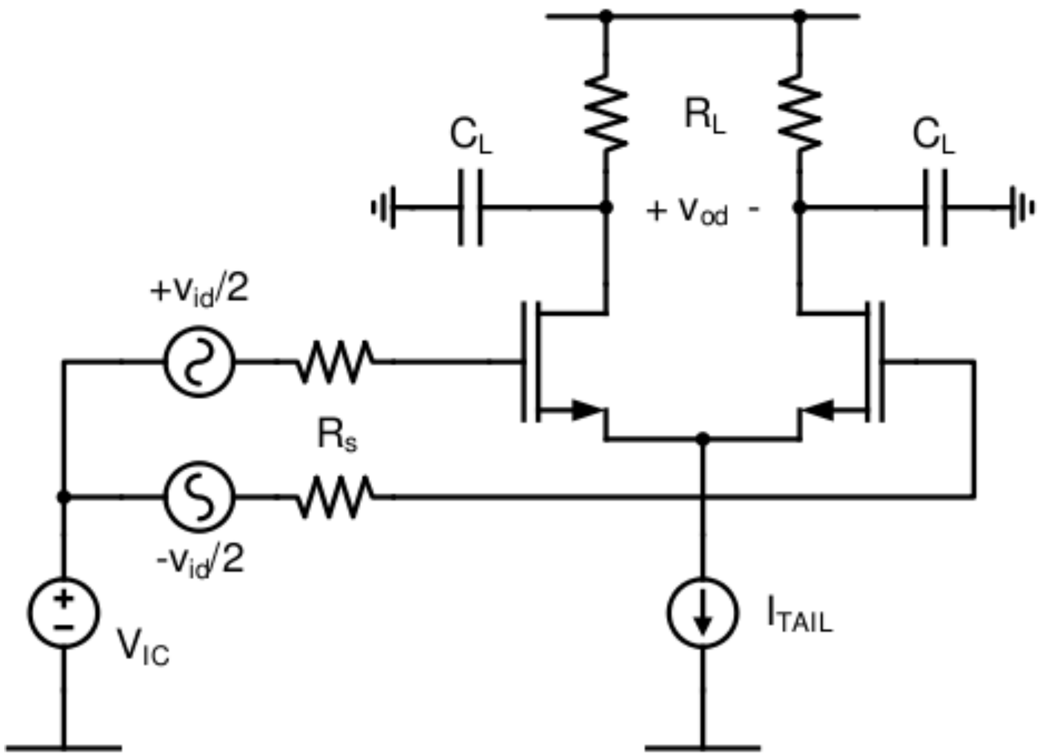
$$R_{in-dm} = \infty$$

$$R_{out-dm} = 2 \cdot (R \,||\, r_o)$$

Observations

- Obtained same amount of gain as a CS stage
- Used twice as many "components" as we used for a CS stage (not a big deal since we plan to build the amplifier inside an IC)
- Used twice as much current
  - To get the same g_m a CS stage need to spend only $I_{TAIL}/2$
- The use of the tail current source makes $I_{D1}+I_{D2}$ independent of V_{IC}
  - The role of the tail current source is to suppress the effect of input common level variations on the operation of M_1 and M_2 and the output levels ($V_{O1} = V_{O2} = V_{DD} - R*I_{TAIL}/2$)
  - The circuit rejects input CM variations (the node V_X absorbs the changes !)
- In the case the output is taken between the two drains, the resulting differential output is entirely a signal component (0 DC component → $V_{OD} = 0$ V)
- The differential pair has high immunity to noise and fluctuations

Differential Pair: Design Example



- Specifications:
  - 0.18 μm technology
  - Av0 = −4
  - RL=1KΩ
  - RS=10KΩ
  - CL=50fF
  - ITAIL ≤ 600μA
  - (VIC = 1V)
- Maximize BW

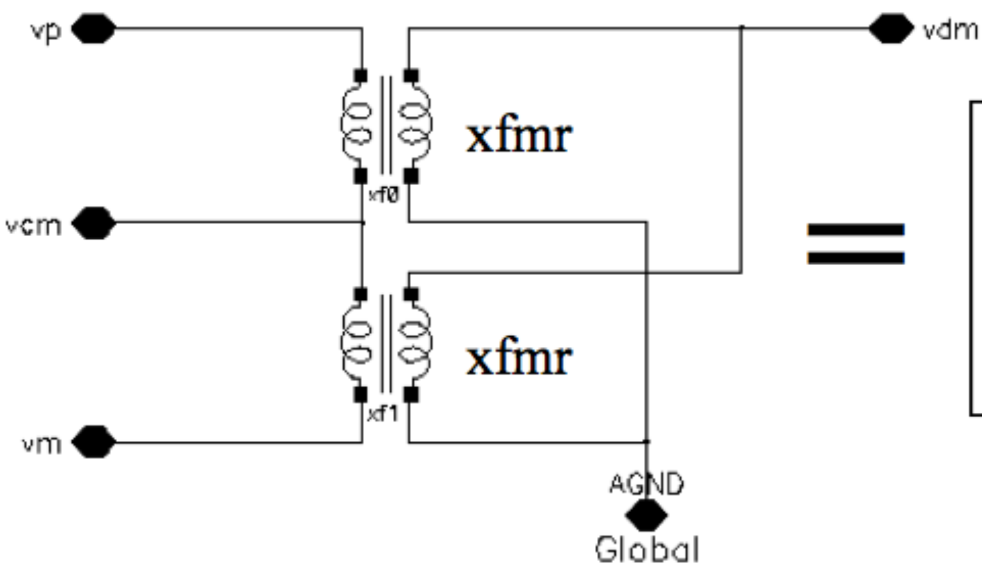
MATLAB design script – 180nm technology

```
%
% gm/ID design of diff pair
% file: design_gmid_dp.m
%
clear all; close all;
addpath('~/gmid_starterkit_2014');
load 180n.mat;

% Specs.
Av0 = 4; RL=1e3; CL=50e-15; RS=10e3; IT=600e-6;
VDD=1.8;
% DC spec.
VIC  = 1

% Calculations
VOUT = VDD - RL*IT/2
gm = Av0/RL
gm_id = gm/(IT/2)
wT = lookup(nch, 'GM_CGG', 'GM_ID', gm_id)
cgd_cgg = lookup(nch, 'CGD_CGG', 'GM_ID', gm_id)
cdd_cgg = lookup(nch, 'CDD_CGG', 'GM_ID', gm_id)
cgg = gm/wT
cgd = cgd_cgg*cgg
cdd = cdd_cgg*cgg
cdb = cdd - cgd
cgs = cgg - cgd

% pole calulations
b1 = RL*CL + cgs*RS + cgd*(RS+RL+gm*RL*RS)
% b1 = coeff of s in den. of Av(s)
fp1 = 1/b1/2/pi
b2 = RS*RL*(cgs*CL + cgs*cgd + CL*cgd)
% b2 = coeff of s^2 in den. of Av(s)
fp2 = 1/2/pi*b1/b2

% device sizing
id_w = lookup(nch, 'ID_W', 'GM_ID', gm_id)
W = IT/2 / id_w
```

```
%
% tail design sizing
%

% Relevant Equations:
% Vx = VIC – VTH – VOV*sqrt(1-0.25*(Vid/VOV)^2)
% VX = VIC – VTH – VOV (= VIC – VGS)

% Lookup VGS with unknown source voltage
% transistor sits on the tail node
VDB1 = VOUT;
VGB1 = VIC;
VGS1 = lookupVGS(nch,'GM_ID',gm_id,'VDB',VDB1, 'VGB', VGB1)
VX = VIC - VGS1
VDS1 = VOUT - VX

LT   = 0.36e-6;
VDST = VX
VSBT = 0;
% pick gm_id such that vod-vdsat = 10%*vod
for gm_id = 5:0.5:20
  VGST = lookupVGS(nch, 'GM_ID', gm_id, 'VDS', VDST, 'VSB',
VSBT, 'L', LT);
  VTHT= lookup(nch, 'VT', 'VGS', VGST, 'VDS', VDST, 'VSB',
VSBT, 'L', LT);
  VODT = VGST - VTHT;
  VDSATT = lookup(nch, 'VDSAT', 'VGS', VGST, 'VDS', VDST,
'VSB', VSBT, 'L', LT);
  ERROR = VODT - VDSATT;
  if ERROR <= 0.1*VODT
    ERROR
    gm_idT = gm_id - 0.5
    break
  end
end

VGST = lookupVGS(nch, 'GM_ID', gm_idT, 'VDS', VDST, 'VSB',
VSBT, 'L', LT)
id_wT = lookup(nch, 'ID_W', 'GM_ID', gm_idT, 'L', LT)
WT = IT/id_wT
```

Aside: Ideal Balun

- Useful for separating CM and DM signal components
- Bidirectional, preserves port impedance
- Uses ideal, inductor-less transformers that works down to DC

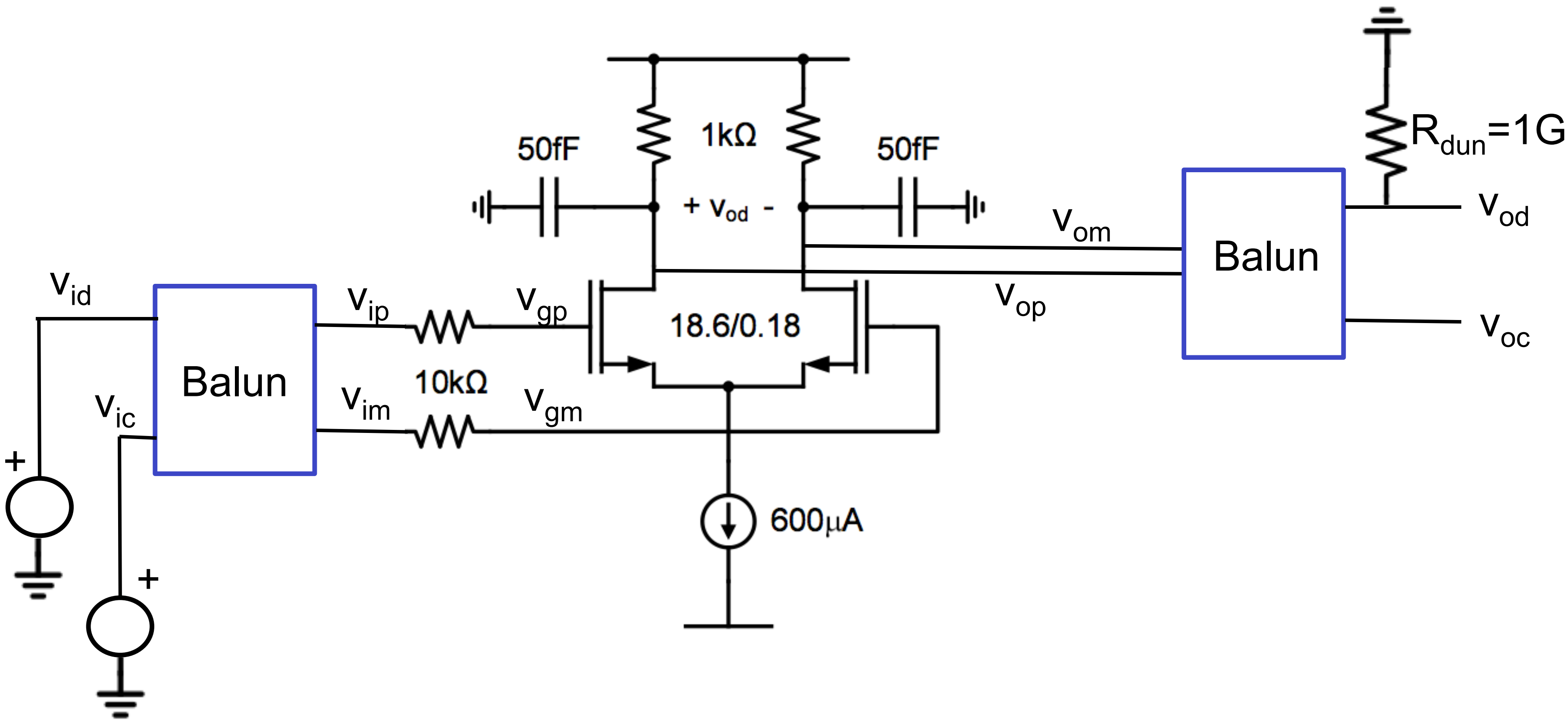


=

```
.subckt balun vdm vcm vp vm
e1 vp vcm transformer vdm 0 2
e2 vcm vm transformer vdm 0 2
.ends balun
```

source: B. Murmann

Circuit



50fF
1kΩ
50fF
+ v_od -
R_dun=1G
v_om
Balun
v_od
v_id
Balun
v_ip
v_gp
18.6/0.18
v_op
v_oc
v_ic
v_im
10kΩ
v_gm
+
+
600μA

Circuit Netlist

```
* gmid_dp.sp

* include models
.include ~/hspice_libs/ee214_hspice.mod

.option post brief accurate

* circuit
vdd vdd 0 1.8
vic vic 0 DC 1
vid vid 0 AC 1
x1 vid vic vip vim balun
x2 vod voc vop vom balun
rdun vod 0 1gig * cannot leave the node floating
it vx 0 600u

m1p vop vgp vx 0 nmos214 W=18.64u L=0.18u
m1m vom vgm vx 0 nmos214 W=18.64u L=0.18u
RLp vdd vop 1K
RLm vdd vom 1K
CLp vop 0 50f
CLm vom 0 50f
RSp vip vgp 10K
RSm vim vgm 10K

* analysis and measurements
.OP
.ac dec 100 1e6 1e12
.pz v(vod) vid

.measure AC GDC find VDB(vod) at 1Meg
.measure AC ADC find V(vod) at 1Meg
.measure AC F3DB when VDB(vod)='GDC - 3'

.alter tail_transistor
vb vb 0 DC 569m
it vx 0 0
MT vx vb 0 0 nmos214 W=150.25u L=0.36u

.end
```

Simulated Operating Point (w/ ideal tail)

```
vx       = 317.6774m

element  0:m1p      0:m1m
 model    0:nmos214  0:nmos214
 region     Saturati   Saturati
  id       300.0000u  300.0000u
  ibs        0.         0.
  ibd        0.         0.
  vgs      682.3226m  682.3226m
  vds        1.1823     1.1823
  vbs     -317.6774m -317.6774m
  vth      564.5355m  564.5355m
  vdsat    109.0106m  109.0106m
  vod      117.7872m  117.7872m
  beta      37.3394m   37.3394m
  gam eff  583.8502m  583.8502m
  gm         4.0749m    4.0749m
  gds      101.0427u  101.0427u
  gmb      887.8890u  887.8890u
  cdtot     20.8733f   20.8733f
  cgtot     37.5590f   37.5590f
  cstot     42.3251f   42.3251f
  cbtot     31.5833f   31.5833f
  cgs       26.8412f   26.8412f
  cgd        8.9865f    8.9865f
```

Good Agreement !

Computed Values:

gm = 4mS

cdd = 22.71 fF

cgg = 37.91 fF

cgd = 8.98 fF

W = 18.66 um

Simulated Operating Point (w/ MOST tail)

```
vx       = 319.5041m


element  0:m1p       0:m1m       0:mt
 model     0:nmos214  0:nmos214  0:nmos214
 region      Saturati    Saturati    Saturati
  id        291.6595u  291.6595u  583.3190u
  ibs         0.          0.          0.
  ibd         0.          0.          0.
  vgs       680.4959m  680.4959m  569.0000m
  vds         1.1888      1.1888    319.5041m
  vbs      -319.5041m -319.5041m     0.
  vth       564.9171m  564.9171m  464.3768m
  vdsat     107.7501m  107.7501m   94.1598m
  vod       115.5789m  115.5789m  104.6232m
  beta       37.3394m   37.3394m  132.1440m
  gam eff   583.8442m  583.8442m  585.5913m
  gm          4.0091m    4.0091m    9.6468m
  gds        99.0495u   99.0495u  151.6471u
  gmb       873.4055u  873.4055u    2.6084m
  cdtot      20.8607f   20.8607f  190.2633f
  cgtot      37.5273f   37.5273f  487.0294f
  cstot      42.2759f   42.2759f  544.7463f
  cbtot      31.5606f   31.5606f  353.6086f
  cgs        26.8026f   26.8026f  385.1031f
  cgd         8.9867f    8.9867f   71.5793f
```

Good Agreement !

Computed Values:

ID1 = 300 uA
IDT = 600 uA

gm1 = 4mS

cdd1 = 22.71 fF
cgg1 = 37.91 fF
cgd1 = 8.98 fF

W1 = 18.66 um

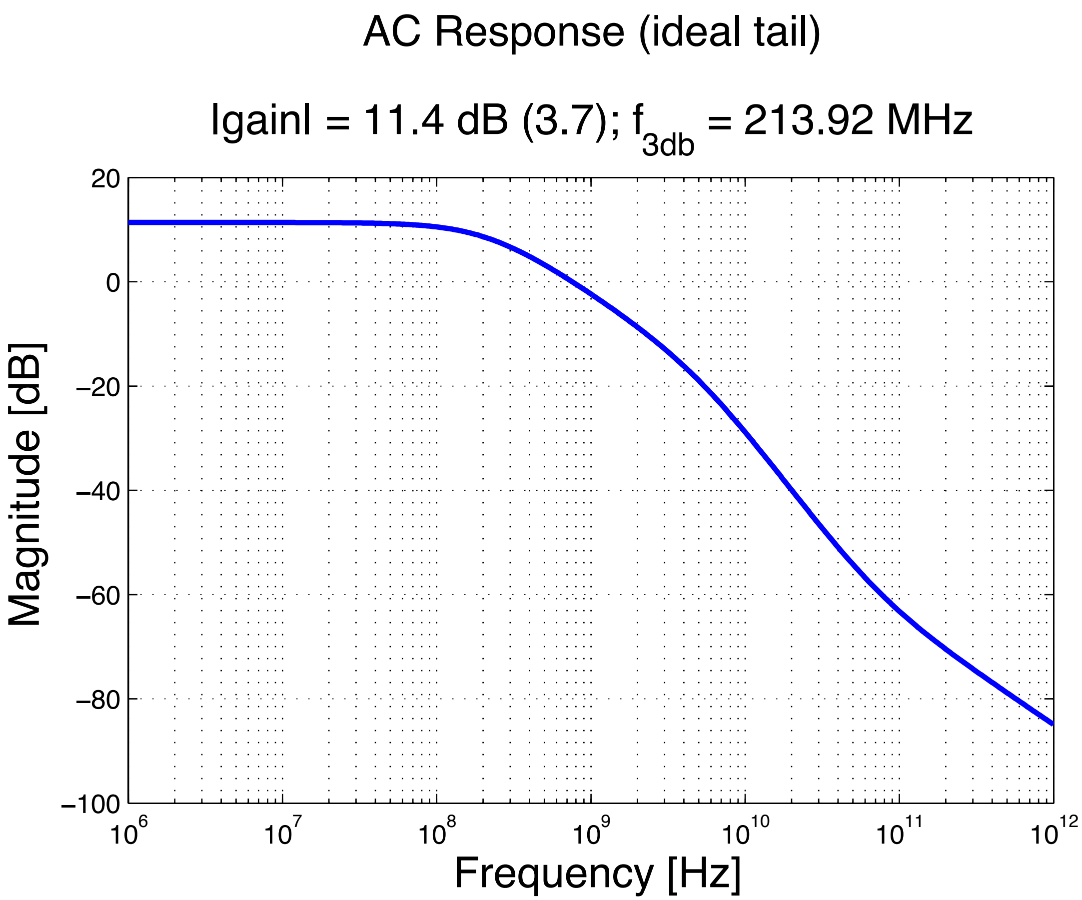
WT = 150.25 um

Plotting HSPICE results in MATLAB

```
%
% file: plot_gmid_dp.m
%
clear all; close all;
format short eng

addpath('/usr/local/MATLAB/HspiceToolbox');
y = loadsig('./spiceout/gmid_dp.ac0');
lssig(y)

figure(1);
freq = evalsig(y,'HERTZ');
vod = evalsig(y,'v_vod');
magdb = 20*log10(abs(vod));
phase = 180*unwrap(angle(vod))/pi;
semilogx(freq, magdb,'linewidth',2, 'color', 'b',
'linestyle', '-');
grid on;
ylabel(' Magnitude [dB]', 'Fontsize', 16);
xlabel(' Frequency [Hz]', 'Fontsize', 16);
xmin = 1e6;
xmax = 1e12;
xlim([xmin xmax]);
av0 = abs(vod(1))
av0db = magdb(1)
f3db = interp1(magdb, freq, magdb(1)-3, 'spline')
% Annotate title
str1 = sprintf('AC Response (ideal tail)\n');
str2 = sprintf('|gain| = %0.3g dB (%0.3g); f_{3db}
= %3.2f MHz', av0db, av0, f3db*1e-6);
str = {str1, str2};
title(str, 'fontsize', 16);
```

```
x = loadsig('./spiceout/gmid_dp.ac1');
lssig(x)

figure(2);
freq = evalsig(x,'HERTZ');
vod = evalsig(x,'v_vod');
magdb = 20*log10(abs(vod));
phase = 180*unwrap(angle(vod))/pi;
semilogx(freq, magdb,'linewidth',2, 'color', 'b',
'linestyle', '-');
grid on;
ylabel(' Magnitude [dB]', 'Fontsize', 16);
xlabel(' Frequency [Hz]', 'Fontsize', 16);
xmin = 1e6;
xmax = 1e12;
xlim([xmin xmax]);
av0 = abs(vod(1))
av0db = magdb(1)
f3db = interp1(magdb, freq, magdb(1)-3, 'spline')
% Annotate title
str1 = sprintf('AC Response (with tail transistor)
\n');
str2 = sprintf('|gain| = %0.3g dB (%0.3g); f_{3db}
= %3.2f MHz', av0db, av0, f3db*1e-6);
str = {str1, str2};
title(str, 'fontsize', 16);
```

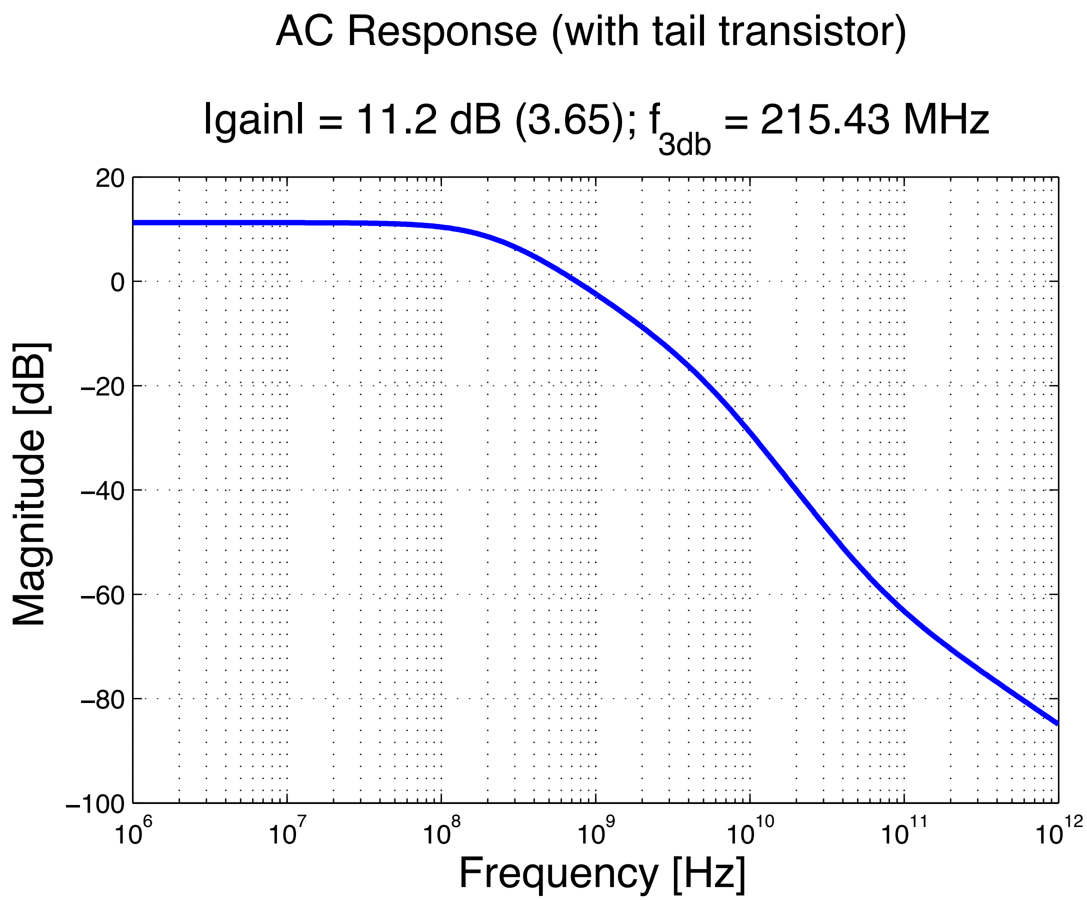
Simulated AC Response (w/ ideal tail)




```
poles ( hertz)
 real                imag
-214.807x           0.
-5.03309g           0.
```

- Calculated values:
  |gain|=12 dB (=4); fp1=199.66 MHz; fp2=5.89GHz

Simulated AC Response (w/ MOST tail)

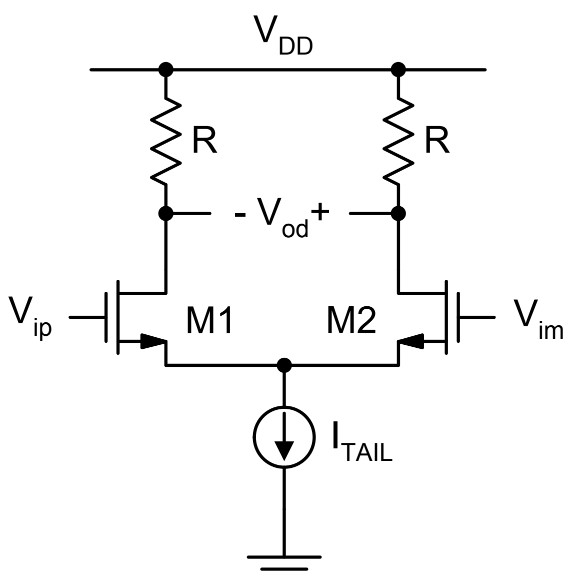



```
poles ( hertz)
real              imag
-216.335x         0.
-4.99376g         0.
```

- Calculated values:
  |gain|=12 dB (=4); fp1=199.66 MHz; fp2=5.89GHz

Voltage Amplifier Transfer Functions

NOTE: $V_{od} = V_{o2} - V_{o1}$



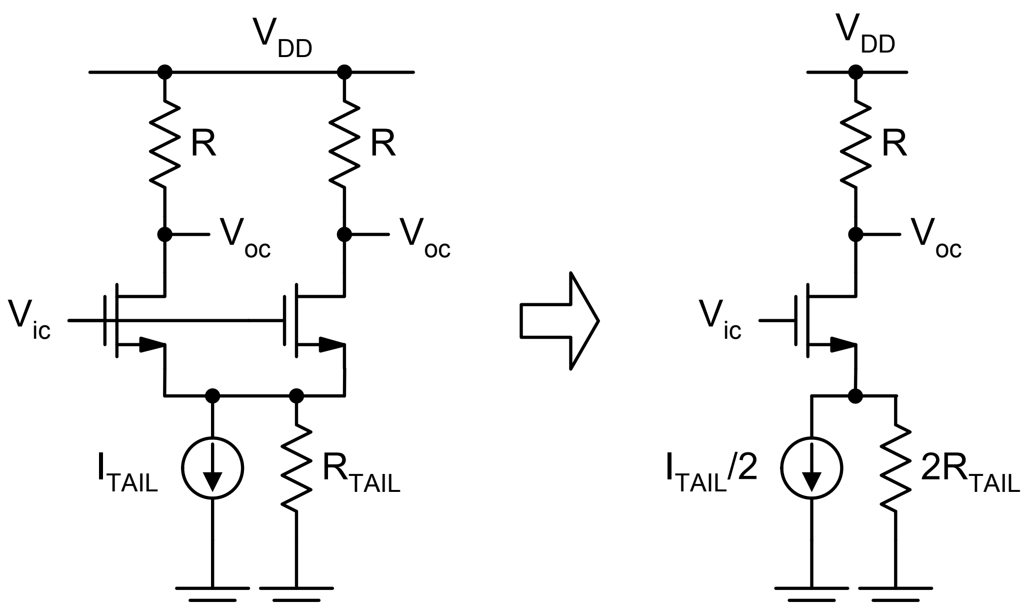
- In a differential amplifier, we primarily want to have large gain that links only the two differential variables

$$A_{dm} = \frac{v_{od}}{v_{id}}$$

- Unfortunately, circuit non-idealities will also cause nonzero "parasitic" gain terms

$$A_{cm} = \frac{v_{oc}}{v_{ic}} \qquad A_{dm-cm} = \frac{v_{oc}}{v_{id}} \qquad A_{cm-dm} = \frac{v_{od}}{v_{ic}}$$

Common Mode Gain



$$A_{cm} = \frac{v_{oc}}{v_{ic}}$$

- Ideally zero ($R_{TAIL}=\infty$)
- With finite R_{TAIL}: $$A_{cm} \simeq -\frac{g_m}{1+g'_m \cdot 2R_{TAIL}}\left(R \parallel r_o(1+g'_m 2R_{TAIL})\right) \approx -\frac{g_m R}{1+g'_m \cdot 2R_{TAIL}}$$

Since degeneration reduces the effective transconductance, and since degeneration occurs only in the CM case, the differential pair is more sensitive to differential inputs than to common-mode inputs ($|A_{cm}| < |A_{dm}|$)

Common Mode Rejection Ratio

- Figure of merit that quantifies ratio of desired/undesired gain
  - Ideally infinite

$$CMRR \triangleq \left| \frac{A_{dm}}{A_{cm}} \right|$$

- For our simple resistively loaded differential pair, this becomes (assuming R << r_o and ignoring body effect)

$$CMRR = \left| \frac{A_{dm}}{A_{cm}} \right| \cong \frac{g_m \cdot R}{\frac{g_m}{1 + g_m \cdot 2R_{TAIL}} \cdot R} = 1 + g_m \cdot 2R_{TAIL}$$

- Other important figures of merit

$$\left| \frac{A_{dm}}{A_{cm-dm}} \right| \qquad \left| \frac{A_{dm}}{A_{dm-cm}} \right|$$

Common Mode Rejection Ratio

- Considering mismatches:

NOTE:
when considering mismatches the circuit become asymmetric; half-circuit cannot be used to analyze the effect of CM fluctuations on the output

$$\left|\frac{A_{dm}}{A_{cm-dm}}\right| \cong \frac{1+g_m \cdot 2R_{TAIL}}{\dfrac{|\Delta g_m|}{g_m}+\dfrac{|\Delta R|}{R}}$$

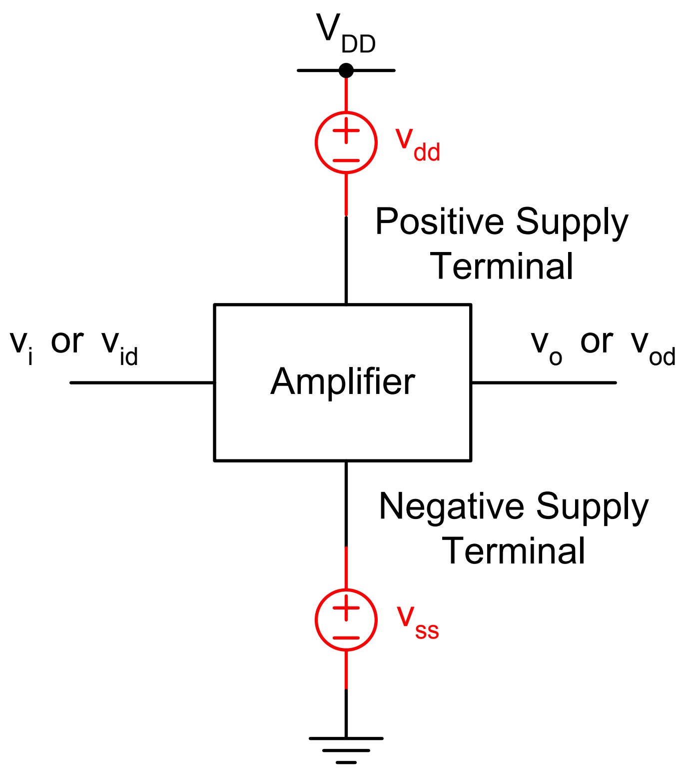
$$g_m \triangleq \frac{g_{m1}+g_{m2}}{2};\ \Delta g_m = g_{m1}-g_{m2} \Leftrightarrow g_{m1} = g_m + \frac{\Delta g_m}{2};\ g_{m2} = g_m - \frac{\Delta g_m}{2}$$

$$R_1 = R$$

$$R_2 = R + \Delta R$$

- Ratio between desired gain and undesired gain can be improved by
  - matching well the transistors and the drain resistances
  - increasing the output resistance of the tail current
- Best option: increase R_{TAIL}
  - Use a large tail transistor
  - Use a cascode for the tail current source

Power Supply Rejection Ratio

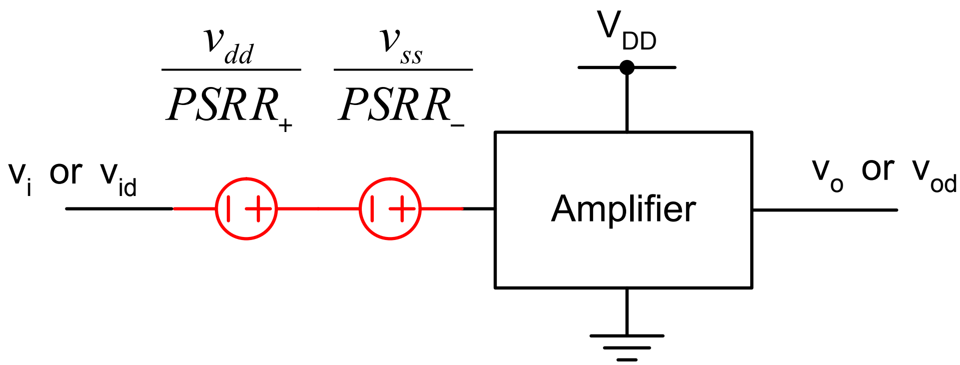


- In practice, "noise" on the supplies will also propagate to the output
  - In a differential system usually due to (half-) circuit imbalance
- Define

$$A_+ = \frac{v_{od}}{v_{dd}} \qquad A_- = \frac{v_{od}}{v_{ss}}$$

$$PSRR_+ = \left|\frac{A_{dm}}{A_+}\right| \qquad PSRR_- = \left|\frac{A_{dm}}{A_-}\right|$$

Input Referred Interpretation

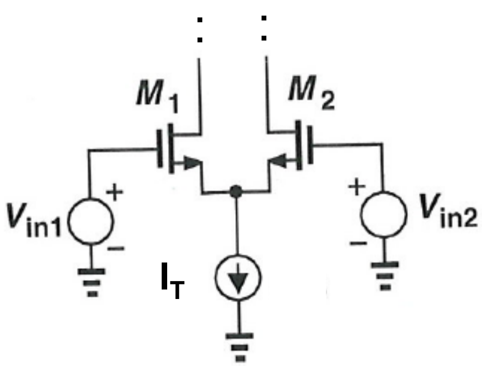


- E.g. 1mV input signal, 100mV supply noise
  - Need PSRR >> 100 (40dB)
- PSRR can be a very critical issue in highly integrated, complex integrated circuits
  - Lots of potential supply noise sources
    - E.g. cross-talk between analog and digital sections

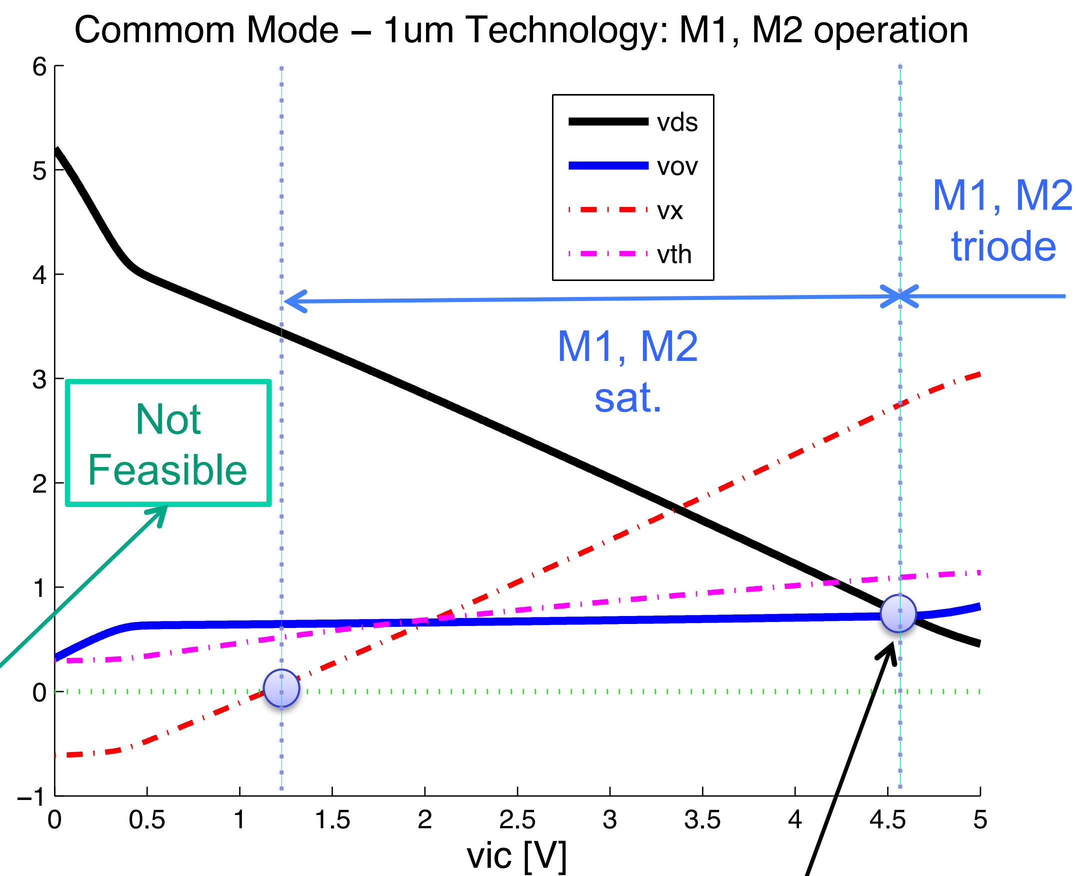
I/O DC characteristics (w/ ideal tail) (1)

Common Mode:

V_{DD} = 5V; I_T=600μA;
W/L=21.33μm/1μm;
R_L=5KΩ



Math "artifact":
Because of the ideal tail the voltage V_X across the tail can take any value (even negative values)

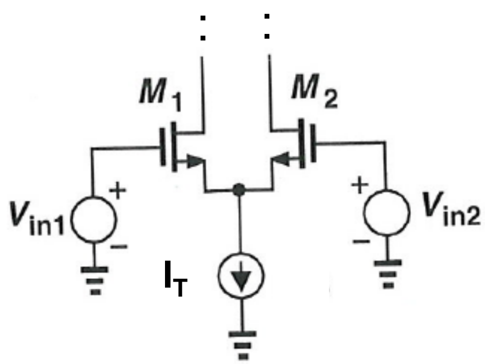


- M_1 and M_2 end up in triode for $V_{IC}(max) \approx V_{DD}+V_{TH}-R_L I_{TAIL}/2 \approx 4.6$ V

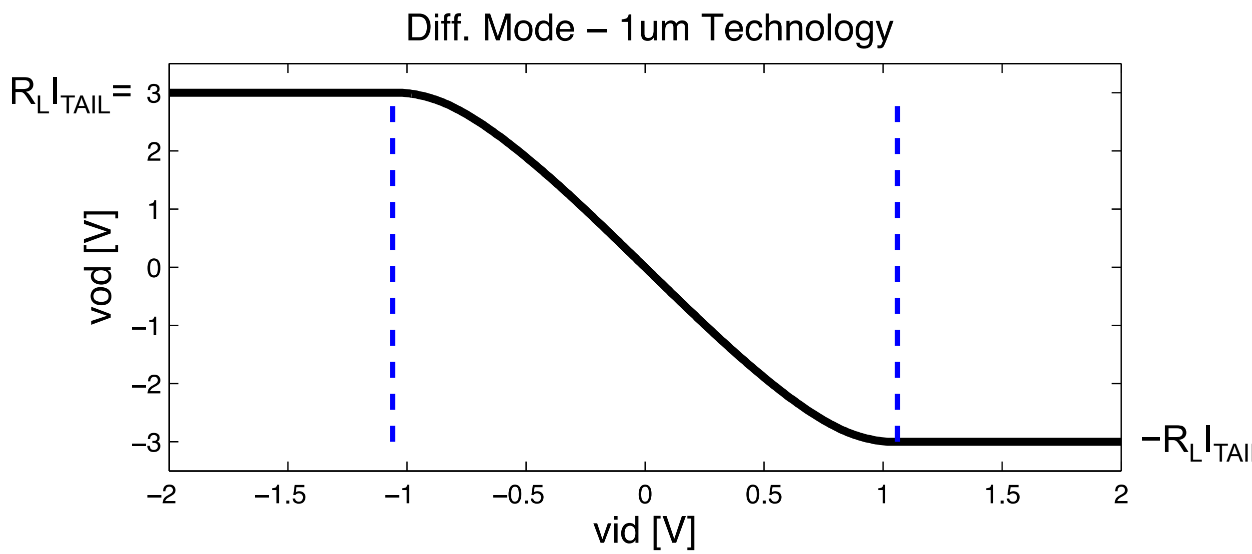
I/O DC characteristics (w/ ideal tail) (2)

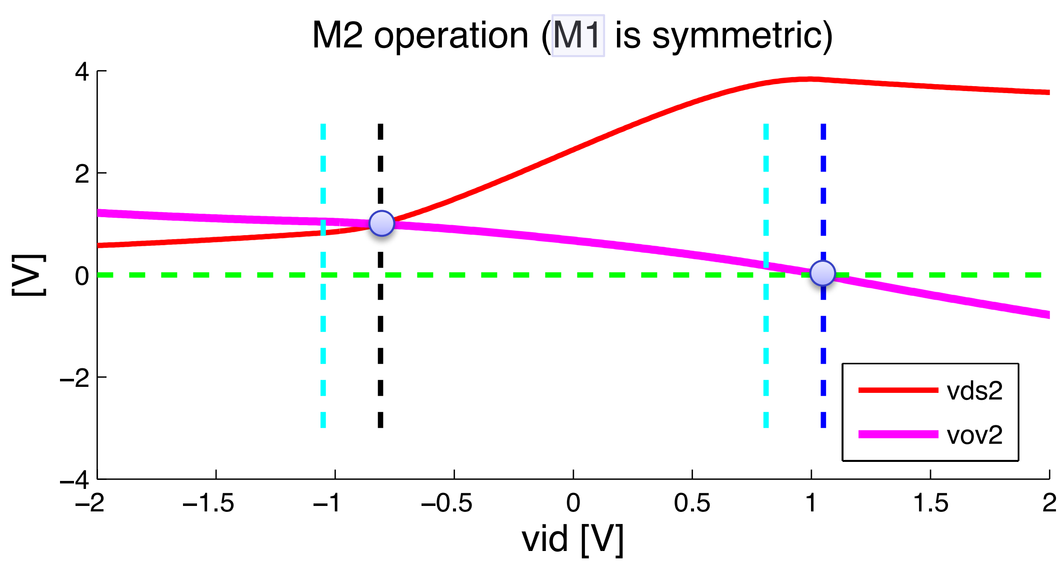
Differential Mode:

V_{DD} = 5V; I_T=600μA;
W/L=21.33μm/1μm;
R_L=5KΩ (V_{IC}=2.5)



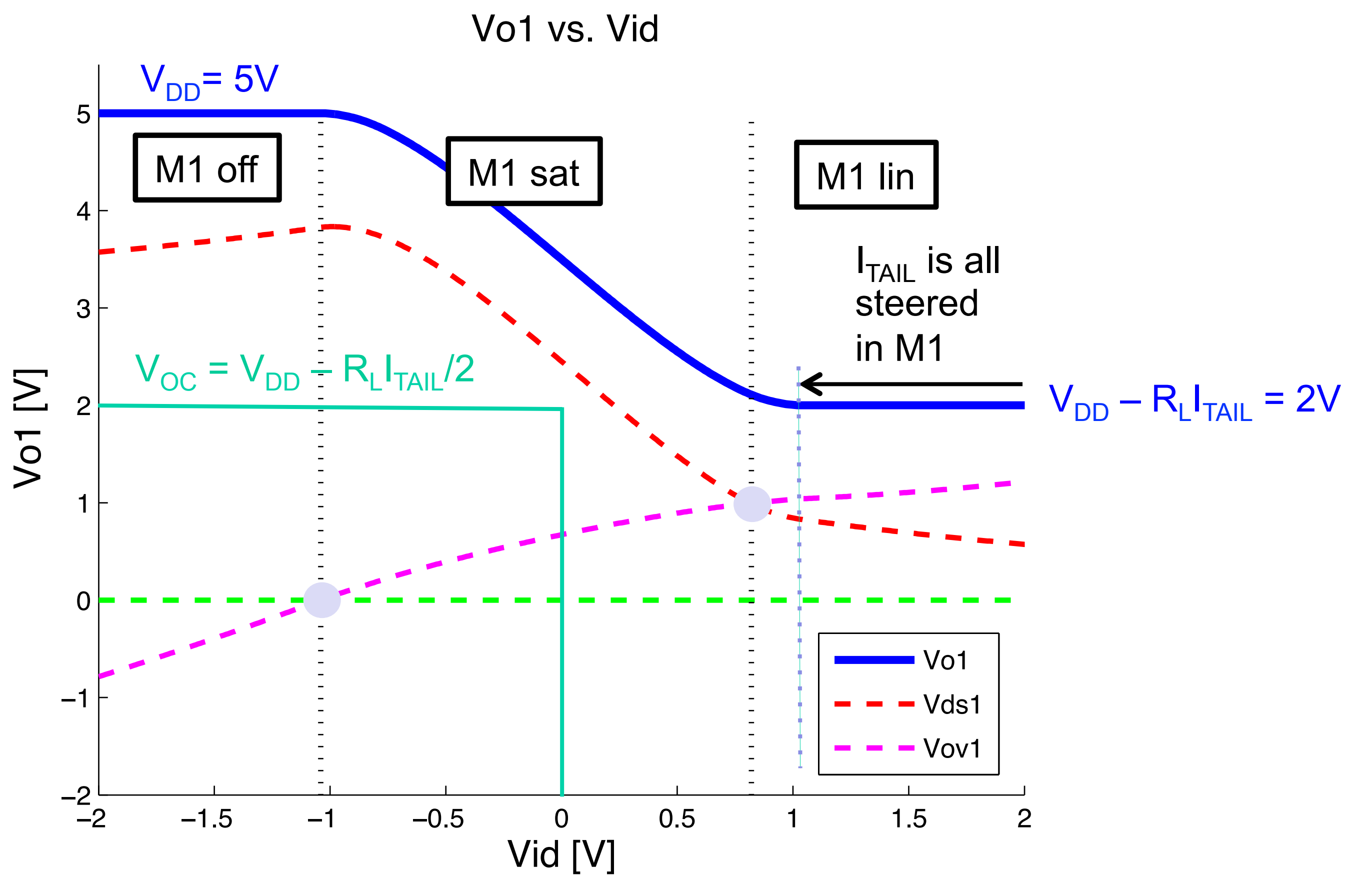
- $V_{id} = V_{in1} - V_{in2}$
- $V_{od} = V_{o1} - V_{o2}$
- $|V_{od}| < R_L I_{TAIL}$





I/O DC characteristics (w/ ideal tail) (3)

output signal swing:



- $V_{o1}(max) = V_{DD}$
- $V_{o1}(min) = V_{OV1} + V_X = V_{IC} - V_{TH} \approx V_{DD} - R_L I_{TAIL}$

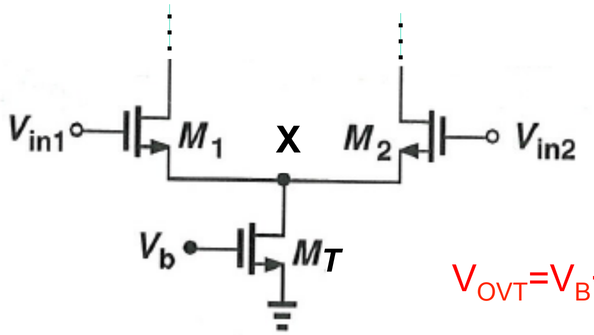
$\Rightarrow$ $| V_{od} | < V_{DD} - V_{IC} + V_{TH} < R_L I_{TAIL}$

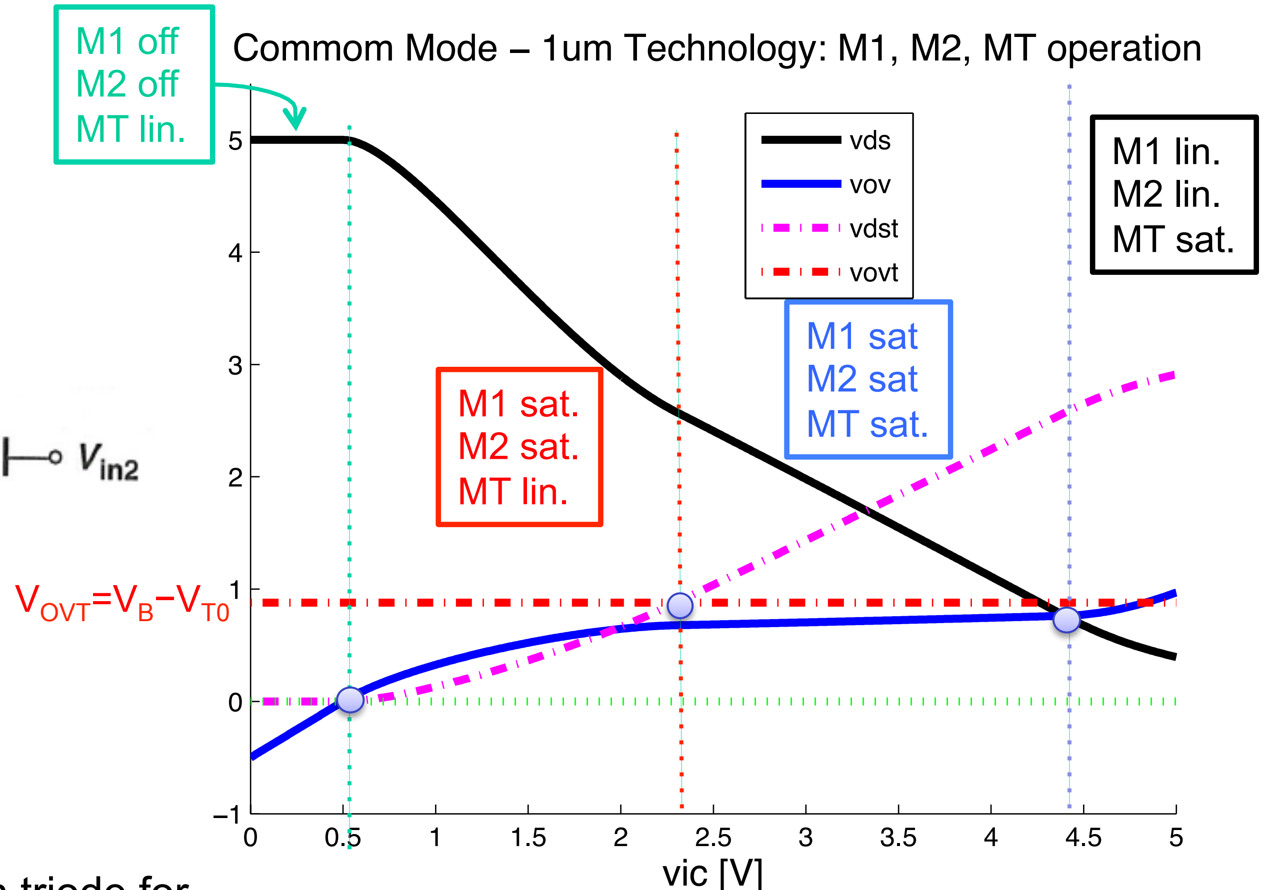
- $V_X = V_{IC} - V_{GS1} = V_{IC} - V_{OV1} - V_{TH}$

I/O DC characteristics (w/ MOST tail) (1)

Common Mode:

V_{DD} = 5V; I_T=600μA;
W/L=21.33μm/1μm;
R_L=5KΩ;
$(W/L)_T$ = 61.07μm/1μm





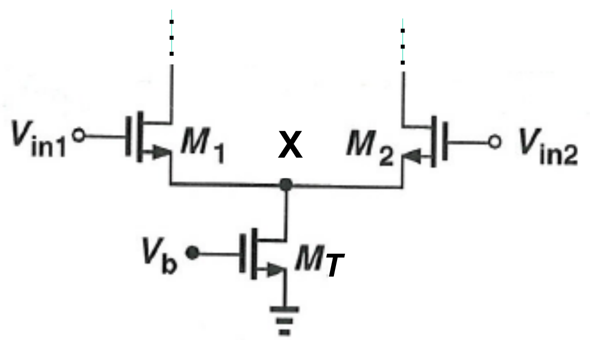
- M_1 and M_2 end up in triode for $V_{IC}(max) \approx V_{DD}+V_{TH}-R_L I_{TAIL}/2 \approx 4.4$ V
- M_T doesn't operate in sat. until $V_{IC}(min) \approx V_{TH}+V_{OV}+V_{OVT} \approx 2.3$V

NOTE:
V_{TH} depends on V_X (a MOST tail provides a "different" V_X than an ideal current source)

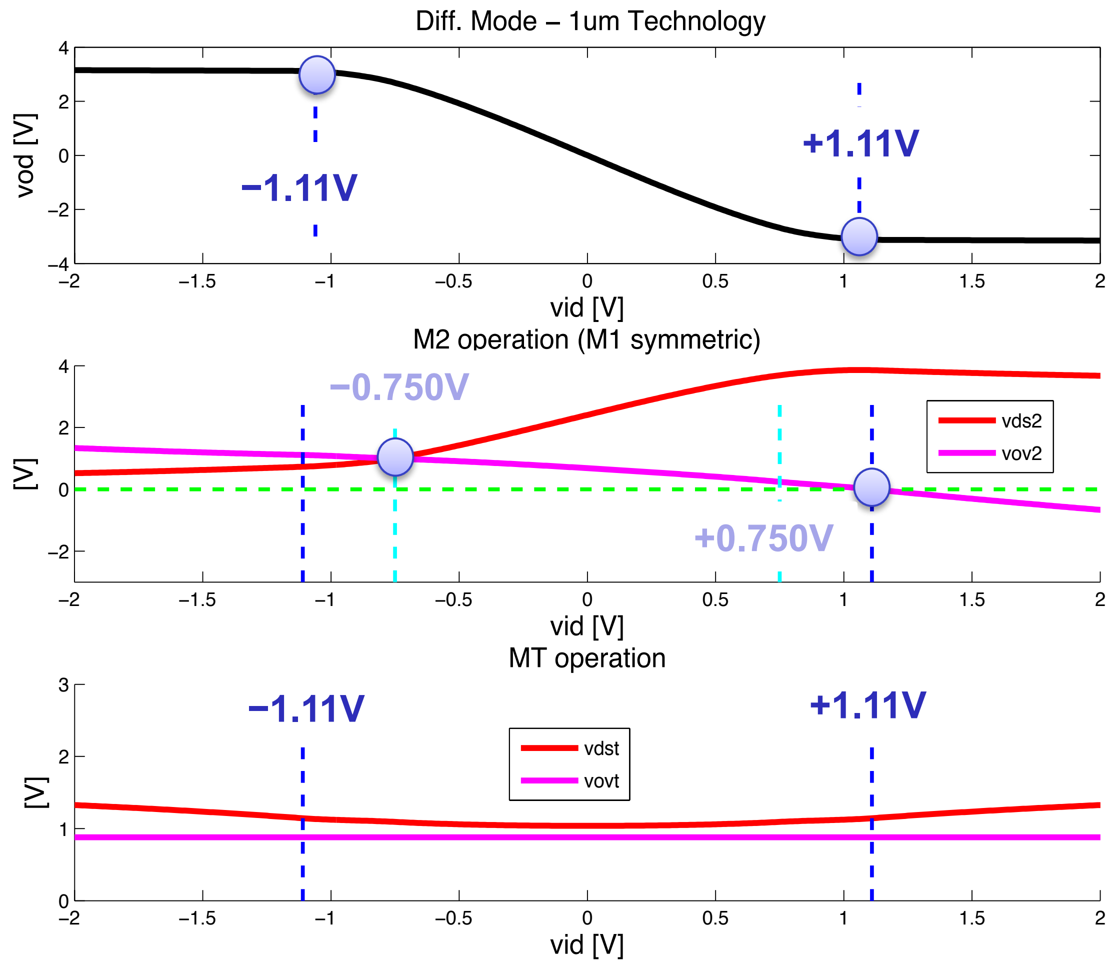
I/O DC characteristics (w/ MOST tail) (2)

Differential Mode:

V_{DD} = 5V; I_T=600µA;
W/L=21.33µm/1µm;
R_L=5KΩ; (VIC=2.5V)
$(W/L)_T$=61.07µm/1µm



NOTE:
The tail MOST operates in sat. for all feasible values of V_{id}
($|V_{id}| < V_{OV,eq}\sqrt{2}$)



MATLAB design script - 1um technology

```
%
% Design of diff pair - 1 um technology
% file: design_dp.m
%
clear all; close all;

% Specs.
Av0 = 4; RL=5e3; CL=50e-15; RS=10e3; IT=600e-6; VDD=5;
% DC spec.
VIC  = 2.5;

% technology
Cox = 2.3e-3; KP=50e-6; L=1e-6; VT0=0.5;
gamma=0.6; PHI=0.8; Ldiff=3e-6; PB=0.95; MJ=0.5;MJSW=0.33;
CJ0 = 0.5e-3; CJSW0=0.5e-9; Cov = 0.5e-9; lambda=0.1;

% Calculations
ID = IT/2;
VOUT = VDD - RL*ID
gm = Av0/RL
VOV = 2*ID/gm
VTH = VT0;     % first pass: neglect body effect
for i=1:10 % iterate a few times
  VGS = VOV + VTH;
  VX = VIC - VGS; % VX = VIC - VTH - VOV (= VIC - VGS)
  VTH = VT0 + gamma*(sqrt(PHI+VX)-sqrt(PHI));
end
VGS
VX
VTH
W = L*2*ID/KP/VOV^2
gm_id = gm/ID
VDS = VOUT - VX

cgd = Cov*W
cgs = Cov*W + 2*Cox*W*L/3
cgg = cgs + cgd;
% neglect cdb and csb

fT = gm/2/pi/cgg
% pole calulations
b1 = RL*CL + cgs*RS + cgd*(RS+RL+gm*RL*RS);
% b1 = coeff of s in den. of Av(s)
fp1 = 1/b1/2/pi
b2 = RS*RL*(cgs*CL + cgs*cgd + CL*cgd);
% b2 = coeff of s^2 in den. of Av(s)
fp2 = 1/2/pi*b1/b2
```

```
%
% tail design sizing
%
LT = 2e-6;
VDST = VX
VOVT_max = VDST % for tail to stay in sat VDST > VOVT
VOVT = 0.9*VOVT_max
WT_LT = 2*IT/VOVT^2/KP
WT = WT_LT*LT
VGST = VOVT + VT0
VB = VGST
```

HSPICE netlist – 1um Technology

```
*
* biasdp.sp
*

* include models
.include ~/hspice_libs/ee114_hspice.mod
.option post brief accurate

* ideal balun
* example instantiation
* x1 vdm vcm vp vm balun
.subckt balun vdm vcm vp vm
e1 vp vcm transformer vdm 0 2
e2 vcm vm transformer vdm 0 2
.ends balun

* circuit
vdd vdd 0 5
vic vic 0 DC 2.5
vid vid 0 AC 1 DC=0

x1 vid vic vip vim balun
x2 vod voc vop vom balun
rdum vod 0 1gig * cannot leave the node floating

it vx 0 600u

m1p vop vgp vx 0 nmos114 W=21.33u L=1u * M1
m1m vom vgm vx 0 nmos114 W=21.33u L=1u * M2
RLp vdd vop 5K
RLm vdd vom 5K
CLp vop 0 50f
CLm vom 0 50f
RSp vip vgp 10k
RSm vim vgm 10k
```

```
* analysis and measurements
.OP
.ac dec 100 1e6 1e12
.pz v(vod) vid
.dc vic -5 5 0.01
.dc vid -5 5 0.01

.probe i(m1m) i(m1p) vth(m1m) vth(m1p) vdsat(m1m) vdsat(m1p)
.probe vgs(m1m) vgs(m1p) vds(m1m) vds(m1p)
.probe vgs(mt) vds(mt) vth(mt) vdsat(mt) i(mt)

.measure AC GDC find VDB(vod) at 1Meg
.measure AC ADC find V(vod) at 1Meg
.measure AC F3DB when VDB(vod)='GDC - 3'

.alter most_tail
it vx 0 0
vb vb 0 1.38
mt vx vb 0 0 nmos114 W=61.07u L=2u

.end
```

Script to plot simulation results (w/ ideal tail) (1)

```
%
% file: plot_biasdp.m
%
clear all; close all;
format short eng

addpath('/usr/local/MATLAB/HspiceToolbox');
x = loadsig('./spiceout/biasdp.sw0');
lssig(x)

vic = evalsig(x,'v_vic');
voc = evalsig(x,'v_voc');
vdsat = evalsig(x,'vdsat_m1m');
vds = evalsig(x,'vds_m1m');
vx = evalsig(x,'v_vx');
vov = evalsig(x,'vgs_m1m') - evalsig(x,'vth_m1m');
vth = evalsig(x,'vth_m1m');

% long channel transistors vov=vdsat
figure(1)
xlim([0 5]);
hold on;
plot(vic,vds,'linewidth',3, 'color', 'k', 'linestyle', '-');
plot(vic,vov,'linewidth',3, 'color', 'b', 'linestyle', '-');
plot(vic,vx,'linewidth',2, 'color', 'r', 'linestyle', '-.');
plot(vic,vth,'linewidth',2, 'color', 'm', 'linestyle', '-.');
% horizontal line across zero
line([0 5], [0 0], 'linestyle', ':', ...
    'linewidth', 2, 'color', 'g');
xlabel('vic [V]', 'Fontsize', 14);
title('Commom Mode - 1um Technology: M1, M2 operation', 'fontsize', 14);
legend('vds', 'vov', 'vx', 'vth', 'location', 'best');

% compute vic_max
for i=1:length(vic)
  if vds(i) < vov(i)
     index = i;
     break
  end
end
% margin: index -1
vic_max_num = vic(index-1)
RL = 5e3; IT=600e-6;vdd = 5; vt0 = 0.5; W = 21.33e-6; L=1e-6; KP=50e-6;
vic_max_an_approx = vdd + vt0 - RL*IT/2 % neglect back gate
vth = vth(index-1)
vic_max_an_precise = vdd + vth - RL*IT/2 % with back gate
```

```
figure(2)
y = loadsig('./spiceout/biasdp.sw1');
lssig(y)

vid = evalsig(y,'v_vid');
vod = evalsig(y,'v_vod');
vdsat = evalsig(y,'vdsat_m1m'); % M2
vds = evalsig(y,'vds_m1m');
vx = evalsig(y,'v_vx');
vgs = evalsig(y,'vgs_m1m');
vth = evalsig(y,'vth_m1m');
vov = vgs - vth;

subplot(2,1,1);
plot(vid,vod,'linewidth',3, 'color', 'k', 'linestyle', '-');
% ideal (no body effect, both MOST in sat)
VOV_eq = sqrt(IT/KP/(W/L))
pointA = sqrt(2)*VOV_eq
pointB = -sqrt(2)*VOV_eq
% ideal vertical lines (no body effect, both MOST in sat)
line([pointA pointA], [-3 3], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'b');
line([pointB pointB], [-3 3], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'b');
xlim([-2 2]);
ylim([-3.5 3.5]);
xlabel('vid [V]', 'Fontsize', 14);
ylabel('vod [V]', 'Fontsize', 14);
title('Diff. Mode - 1um Technology', 'fontsize', 14);

subplot(2,1,2)
hold on;
plot(vid,vds,'linewidth',2, 'color', 'r', 'linestyle', '-');
plot(vid,vov,'linewidth',3, 'color', 'm', 'linestyle', '-');
xlim([-2 2]);
% horizontal line across zero
line([-2 2], [0 0], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'g');
legend('vds2', 'vov2', 'location', 'best');
xlabel('vid [V]', 'Fontsize', 14);
ylabel('[V]', 'Fontsize', 14);
title('M2 operation (M1 is symmetric)', 'fontsize', 14);
```

to be continued …

Script to plot simulation results (w/ ideal tail) (2)

```
% compute M2 operation
for i=1:length(vid)
  if vov(i) < 0
    index = i;
    break
  end
end
M2cutoff = vid(index)
line([M2cutoff M2cutoff], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'b');
for i=1:length(vid)
  if vds(i) > vov(i)
    index = i;
    break
  end
end
M2sat = vid(index)
line([M2sat M2sat], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'k');
% M1 operation (symmetric)
line([-M2cutoff -M2cutoff], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'c');
line([-M2sat -M2sat], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'c');
```

```
figure(3);
hold on;
vout1 = evalsig(y,'v_vop');
vgs1 = evalsig(y,'vgs_m1p');
vth1 = evalsig(y,'vth_m1p');
vds1 = evalsig(y,'vds_m1p');
vov1 = vgs1 - vth1;
plot(vid,vout1,'linewidth',3, 'color', 'b', 'linestyle', '-');
plot(vid,vds1,'linewidth',2, 'color', 'r', 'linestyle', '--');
plot(vid,vov1,'linewidth',2, 'color', 'm', 'linestyle', '--');
% horizontal line across zero
line([-2 2], [0 0], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'g');
xlim([-2 2]);
ylim([-2 5.5]);
xlabel('Vid [V]', 'Fontsize', 14);
ylabel('Vo1 [V]', 'Fontsize', 14);
title('Vo1 vs. Vid', 'fontsize', 14);
legend('Vo1','Vds1','Vov1','location', 'best')
% compute M1 operation
for i=1:length(vid)
  if vov1(i) > 0
    index = i;
    break
  end
end
M1sat = vid(index)
line([M1sat M1sat], [-3 5], 'linestyle', ':', ...
     'linewidth', 2, 'color', 'k');
for i=1:length(vid)
  if vds1(i) < vov1(i)
    index = i;
    break
  end
end
M1triode = vid(index)
line([M1triode M1triode], [-3 5], 'linestyle', ':', ...
     'linewidth', 2, 'color', 'k');
```

Script to plot simulation results (w/ MOST tail) (1)

```
%
% file: plot_biasdp_wtail.m
%
clear all; close all;
format short eng

addpath('/usr/local/MATLAB/HspiceToolbox');
x = loadsig('./spiceout/biasdp.sw2');
lssig(x)

vic = evalsig(x,'v_vic');
voc = evalsig(x,'v_voc');
vgs = evalsig(x,'vgs_m1m');
vds = evalsig(x,'vds_m1m');
vth = evalsig(x,'vth_m1m');
vov = vgs - vth;
vx = evalsig(x,'v_vx');
vgst = evalsig(x,'vgs_mt');
vdst = evalsig(x,'vds_mt');
vtht = evalsig(x,'vth_mt');
vovt = vgst - vtht;

% long channel transistors vov=vdsat
figure(1)
xlim([0 5]);
ylim([-1 5.5]);
hold on;
plot(vic,vds,'linewidth',3, 'color', 'k', 'linestyle', '-');
plot(vic,vov,'linewidth',3, 'color', 'b', 'linestyle', '-');
plot(vic,vdst,'linewidth',3, 'color', 'm', 'linestyle', '-.');
plot(vic,vovt,'linewidth',3, 'color', 'r', 'linestyle', '-.');
% horizontal line across zero
line([0 5], [0 0], 'linestyle', ':', ...
    'linewidth', 3, 'color', 'g');
xlabel('vic [V]', 'Fontsize', 14);
title('Commom Mode - 1um Technology: M1, M2, MT operation', 'fontsize', 14);
legend('vds', 'vov', 'vdst', 'vovt','location', 'best');

% compute vic_max
for i=1:length(vic)
  if vds(i) < vov(i)
     index = i;
     break
  end
end
% margin: index -1
vic_max_num = vic(index-1)
RL = 5e3; IT=600e-6;vdd = 5; vt0 = 0.5; W = 21.33e-6; L=1e-6; KP=50e-6;
vic_max_an_approx = vdd + vt0 - RL*IT/2 % neglect back gate
vth = vth(index-1)
vic_max_an_precise = vdd + vth - RL*IT/2 % with back gate
% compute vic_min
for i=1:length(vic)
  if vdst(i) > vovt(i)
     index = i;
     break
  end
end
vic_min_num = vic(index)
```

```
figure(2)
y = loadsig('./spiceout/biasdp.sw3');
lssig(y)

vid = evalsig(y,'v_vid');
vod = evalsig(y,'v_vod');
vx = evalsig(y,'v_vx');
vdsat = evalsig(y,'vdsat_m1m'); % M2
vds = evalsig(y,'vds_m1m');
vgs = evalsig(y,'vgs_m1m');
vth = evalsig(y,'vth_m1m');
vov = vgs - vth;
vgst = evalsig(y,'vgs_mt');
vdst = evalsig(y,'vds_mt');
vtht = evalsig(y,'vth_mt');
vovt = vgst - vtht;

subplot(3,1,1);
plot(vid,vod,'linewidth',3, 'color', 'k', 'linestyle', '-');
% ideal (no body effect, both MOST in sat)
VOV_eq = sqrt(IT/KP/(W/L))
pointA = sqrt(2)*VOV_eq
pointB = -sqrt(2)*VOV_eq
% ideal vertical lines (no body effect, both MOST in sat)
line([pointA pointA], [-3 3], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'b');
line([pointB pointB], [-3 3], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'b');
xlim([-2 2]);
ylim([-4 4]);
xlabel('vid [V]', 'Fontsize', 14);
ylabel('vod [V]', 'Fontsize', 14);
title('Diff. Mode - 1um Technology', 'fontsize', 14);
```

to be continued …

Script to plot simulation results (w/ MOST tail) (2)

```
subplot(3,1,2)
hold on;
plot(vid,vds,'linewidth',3, 'color', 'r', 'linestyle', '-');
plot(vid,vov,'linewidth',3, 'color', 'm', 'linestyle', '-');
xlim([-2 2]);
ylim([-3 4]);
% horizontal line across zero
line([-2 2], [0 0], 'linestyle', '--', ...
    'linewidth', 2, 'color', 'g');
legend('vds2', 'vov2', 'location', 'best');
xlabel('vid [V]', 'Fontsize', 14);
ylabel('[V]', 'Fontsize', 14);
title('M2 operation (M1 symmetric)', 'fontsize', 14);
% compute M2 operation
for i=1:length(vid)
  if vov(i) < 0
    index = i;
    break
  end
end
M2cutoff = vid(index-1)
line([M2cutoff M2cutoff], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'b');
for i=1:length(vid)
  if vds(i) > vov(i)
    index = i;
    break
  end
end
M2sat = vid(index)
line([M2sat M2sat], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'c');
% M1 operation (symmetric)
line([-M2cutoff -M2cutoff], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'b');
line([-M2sat -M2sat], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'c');

subplot(3,1,3)
hold on;
plot(vid,vdst,'linewidth',3, 'color', 'r', 'linestyle', '-');
plot(vid,vovt,'linewidth',3, 'color', 'm', 'linestyle', '-');
xlim([-2 2]);
ylim([0 3]);
legend('vdst', 'vovt','location', 'best');
xlabel('vid [V]', 'Fontsize', 14);
ylabel('[V]', 'Fontsize', 14);
title('MT operation', 'fontsize', 14);
line([-M2cutoff -M2cutoff], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'b');
line([M2cutoff M2cutoff], [-3 3], 'linestyle', '--', ...
     'linewidth', 2, 'color', 'b');
```