EE214B
g_m/I_D-Based Design

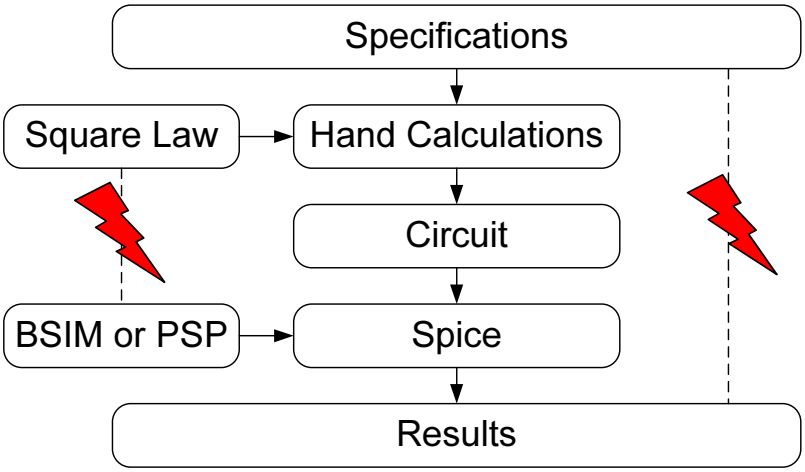
Handout #6

B. Murmann
Stanford University
Winter 2012-13

Summary on MOSFET Modeling

- Modern MOSFETs are complicated!
- The IV-behavior in saturation can be roughly categorized according to the channel's inversion level: weak, moderate and strong inversion
- The current is due to diffusion in weak inversion and mostly due to drift in strong inversion; the transition is smooth and complicated
- The classic square law model is based on an ideal drift model, and applies only near the onset of strong inversion
  - And even then, the predictions are inaccurate unless short channel effects are taken into account
- The bottom line is that there is no modeling expression that is simple enough for hand analysis and sufficiently accurate to match real world device behavior

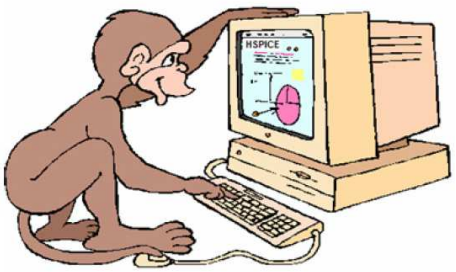
The Problem



- Since there is a disconnect between actual transistor behavior and the simple square law model, any square-law driven design optimization will be far off from Spice results

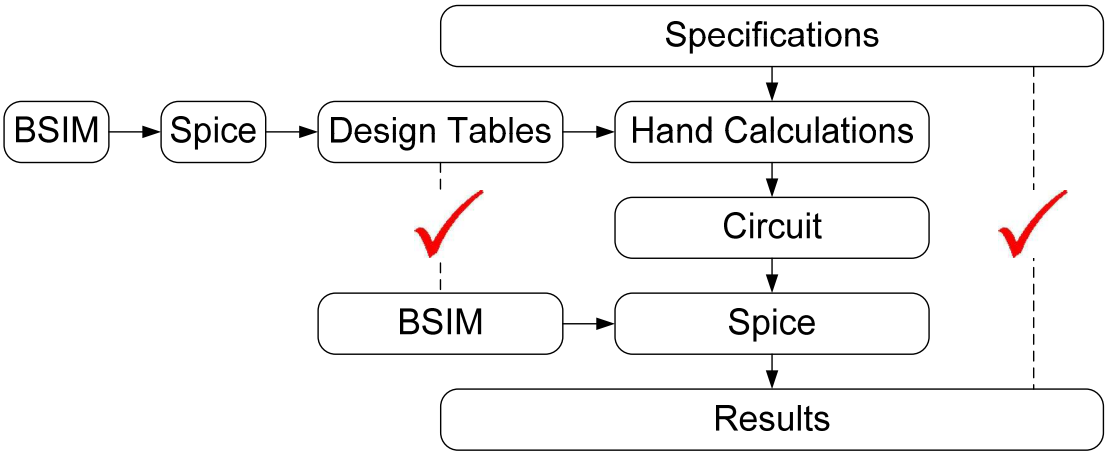
Unfortunate Consequence

- In absence of a simple set of equations for hand analysis, many designers tend to converge toward a "spice monkey" design methodology
  - No hand calculations, iterate in spice until the circuit "somehow" meets the specifications
  - Typically results in sub-optimal designs, uninformed design decisions, etc.
- Our goal
  - Maintain a systematic design methodology in absence of a set of compact MOSFET equations
- Strategy
  - Design using look-up tables or charts



[Courtesy Isaac Martinez]

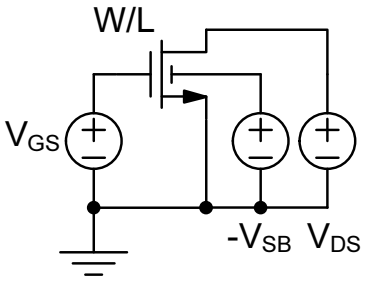
The Solution



- Use pre-computed spice data in hand calculations

Starting Point: Technology Characterization via DC Sweep

```
* /usr/class/ee214b/hspice/techchar.sp

.inc '/usr/class/ee214b/hspice/ee214_hspice.sp'
.inc 'techchar_params.sp'
.param ds = 0.9
.param gs = 0.9

vdsn     vdn 0          dc 'ds'
vgsn     vgn 0          dc 'gs'
vbsn     vbn 0          dc '-subvol'
mn       vdn vgn 0 vbn nmos214  L='length' W='width'

.options dccap post brief accurate nomod
.dc gs 0 'gsmax' 'gsstep' ds 0 'dsmax' 'dsstep'

.probe n_id   = par('i(mn)')
.probe n_vt   = par('vth(mn)')
.probe n_gm   = par('gmo(mn)')
.probe n_gmb  = par('gmbso(mn)')
.probe n_gds  = par('gdso(mn)')
.probe n_cgg  = par('cggbo(mn)')
.probe n_cgs  = par('-cgsbo(mn)')
.probe n_cgd  = par('-cgdbo(mn)')
.probe n_cgb  = par('cbgbo(mn)')
.probe n_cdd  = par('cddbo(mn)')
.probe n_css  = par('cssbo(mn)')
```



Matlab Wrapper

```
% /usr/class/ee214b/hspice/techchar.m

% HSpice toolbox
addpath('/usr/class/ee214b/matlab/hspice_toolbox')

% Parameters for HSpice runs
VGS_step = 25e-3; VDS_step = 25e-3; VS_step = 0.1;
VGS_max = 1.8;    VDS_max = 1.8;    VS_max = 1;
VGS = 0:VGS_step:VGS_max; VDS = 0:VDS_step:VDS_max; VS = 0:VS_step:VS_max;
W = 5; L = [(0.18:0.02:0.5) (0.6:0.1:1.0)];

% HSpice simulation loop
for i = 1:length(L)
  for j = 1:length(VS)
    % write out circuit parameters and run hspice
    fid = fopen('techchar_params.sp', 'w');
    fprintf(fid,'*** simulation parameters **** %s\n', datestr(now));
    fprintf(fid,'.param width  = %d\n', W*1e-6);
    fprintf(fid,'.param length = %d\n', L(i)*1e-6);
    fprintf(fid,'.param subvol = %d\n', VS(j));
    fprintf(fid,'.param gsstep = %d\n', VGS_step);
    fprintf(fid,'.param dsstep = %d\n', VDS_step);
    fprintf(fid,'.param gsmax  = %d\n', VGS_max);
    fprintf(fid,'.param dsmax  = %d\n', VDS_max);
    fclose(fid);

    system('/usr/class/ee/synopsys/hspice/F-2011.09-SP2/hspice/bin/hspice techchar.sp >!...
            techchar.out');
    end
end
```


Simulation Data in Matlab

```
% data stored in /usr/class/ee214b/matlab
>> load 180nch.mat;

>> nch

nch =
     ID: [4-D double]
     VT: [4-D double]
     GM: [4-D double]
    GMB: [4-D double]
    GDS: [4-D double]
    CGG: [4-D double]
    CGS: [4-D double]
    CGD: [4-D double]
    CGB: [4-D double]
    CDD: [4-D double]
    CSS: [4-D double]
    VGS: [73x1 double]
    VDS: [73x1 double]
     VS: [11x1 double]
      L: [22x1 double]
      W: 5

>> size(nch.ID)

ans =
    22    73    73    11
```

Four-dimensional arrays

$$I_D(L, V_{GS}, V_{DS}, V_S)$$

$$V_t(L, V_{GS}, V_{DS}, V_S)$$

$$g_m(L, V_{GS}, V_{DS}, V_S)$$

…

Lookup Function (For Convenience)

```
>> lookup(nch, 'ID', 'VGS', 0.5, 'VDS', 0.5)

ans =
  8.4181e-006

>> help lookup

  The function "lookup" extracts a desired subset from the 4-dimensional
  simulation data. The function interpolates when the requested points lie off
  the simulation grid.

    There are three basic usage modes:
    (1) Simple lookup of parameters at given (L, VGS, VDS, VS)
    (2) Lookup of arbitrary ratios of parameters, e.g. GM_ID, GM_CGG at given
        (L, VGS, VDS, VS)
    (3) Cross-lookup of one ratio against another, e.g. GM_CGG for some GM_ID

    In usage scenarios (1) and (2) the input parameters (L, VGS, VDS, VS) can be
    listed in any order and default to the following values when not specified:

    L = min(data.L); (minimum length used in simulation)
    VGS = data.VGS; (VGS vector used during simulation)
    VDS = max(data.VDS)/2; (VDD/2)
    VS = 0;
```


Key Question

- How can we use all this data for **systematic** design?
- Many options exist
  - And you can invent your own, if you like
- Method taught in EE214B
  - Look at the transistor in terms of width-independent figures of merit that are intimately linked to design specification (rather than some physical modeling parameters that do not directly relate to circuit specs)
  - Think about the design tradeoffs in terms of the MOSFET's inversion level, using g_m/I_D as a proxy

Figures of Merit for Design

Square Law

- Transconductance efficiency
  - Want large g_m, for as little current as possible

$$\frac{g_m}{I_D} = \frac{2}{V_{OV}}$$

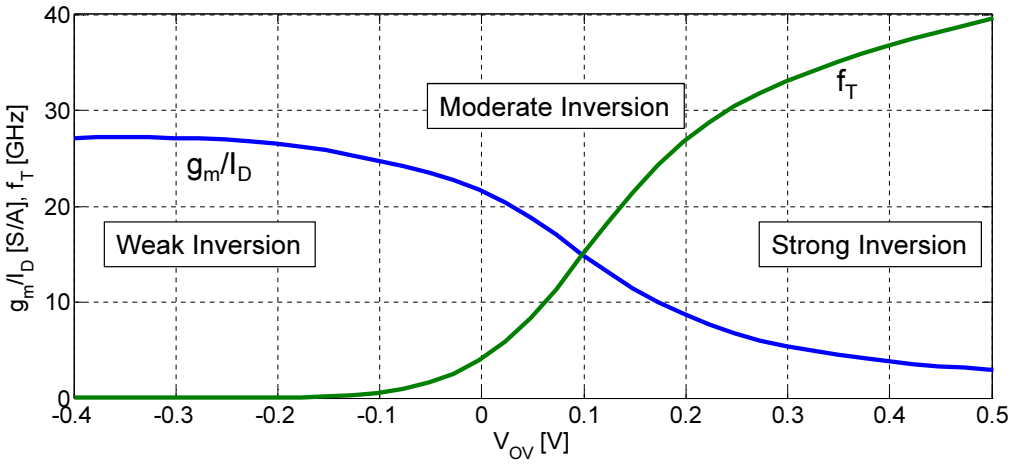
- Transit frequency
  - Want large g_m, without large C_{gg}

$$\frac{g_m}{C_{gg}} \cong \frac{3}{2}\frac{\mu V_{OV}}{L^2}$$

- Intrinsic gain
  - Want large g_m, but no g_o

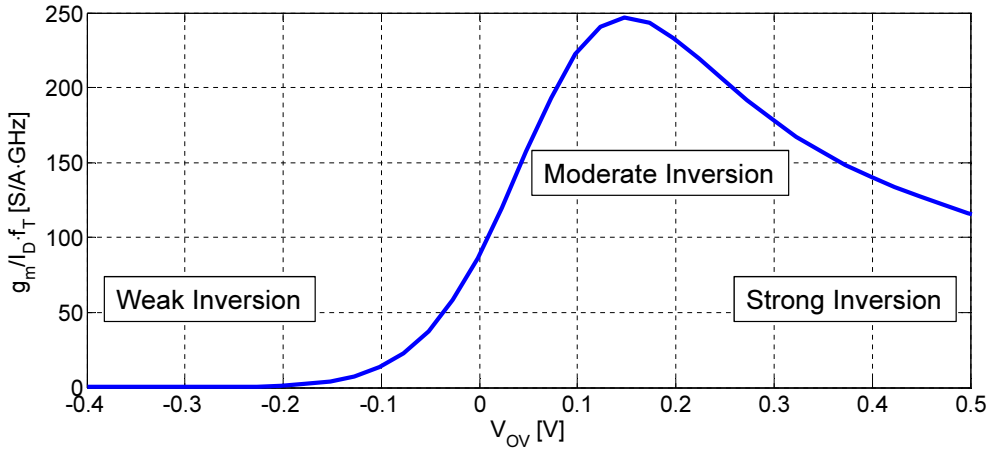
$$\frac{g_m}{g_o} \cong \frac{2}{\lambda V_{OV}}$$

Design Tradeoff: g_m/I_D and f_T



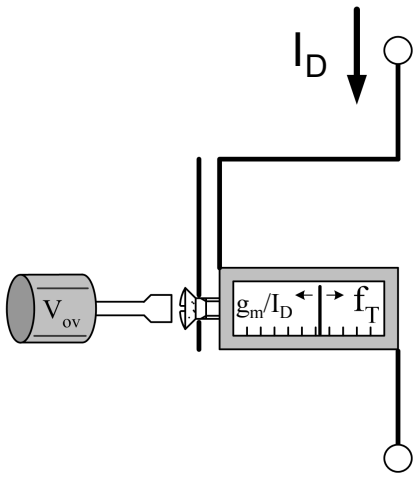
- Weak inversion: Large g_m/ID (>20 S/A), but small f_T
- Strong inversion: Small g_m/ID (<10 S/A), but large f_T

Product of g_m/I_D and f_T



- Interestingly, the product of g_m/I_D and f_T peaks in moderate inversion
- Operating the transistor in moderate inversion is optimal when we value speed and power efficiency equally
  - Not always the case

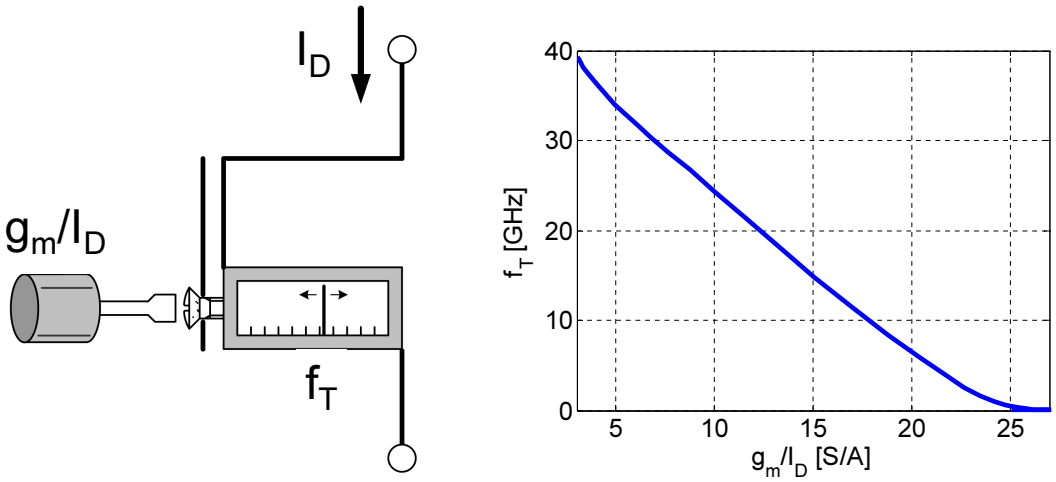
Design in a Nutshell



- Choose the inversion level according to the proper tradeoff between speed (f_T) and efficiency (g_m/I_D) for the given circuit
- The inversion level is fully determined by the gate overdrive V_{OV}
  - But, V_{OV} is not a very interesting parameter outside the square law framework; not much can be computed from V_{OV}

Eliminating V_{OV}

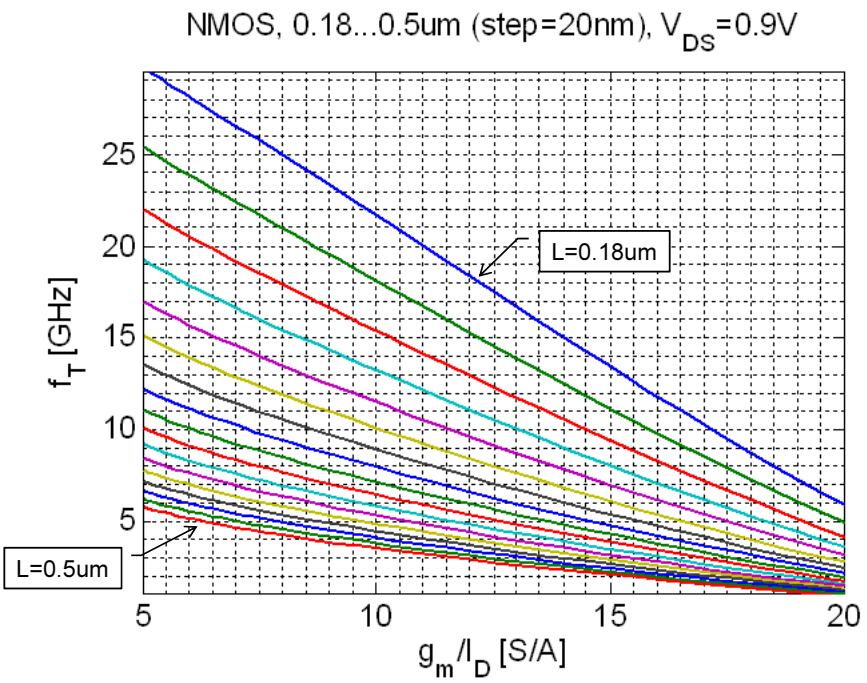
- The inversion level is also fully defined once we pick g_m/I_D, so there is no need to know V_{OV}



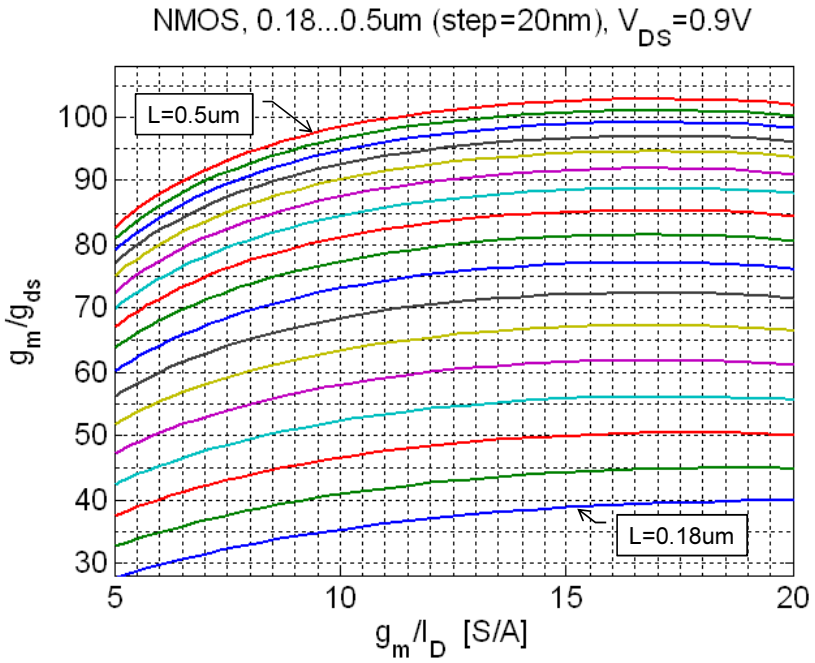
g_m/I_D-centric Technology Characterization

- Tabulate the following parameters for a reasonable range of g_m/I_D and channel lengths
  - Transit frequency (f_T)
  - Intrinsic gain (g_m/g_o)
- Also tabulate relative estimates of extrinsic capacitances
  - C_{gd}/C_{gg} and C_{dd}/C_{gg}
- Note that all of these parameters are (to first order) independent of device width
- In order to compute device widths, we need one more table that links g_m/I_D and current density I_D/W

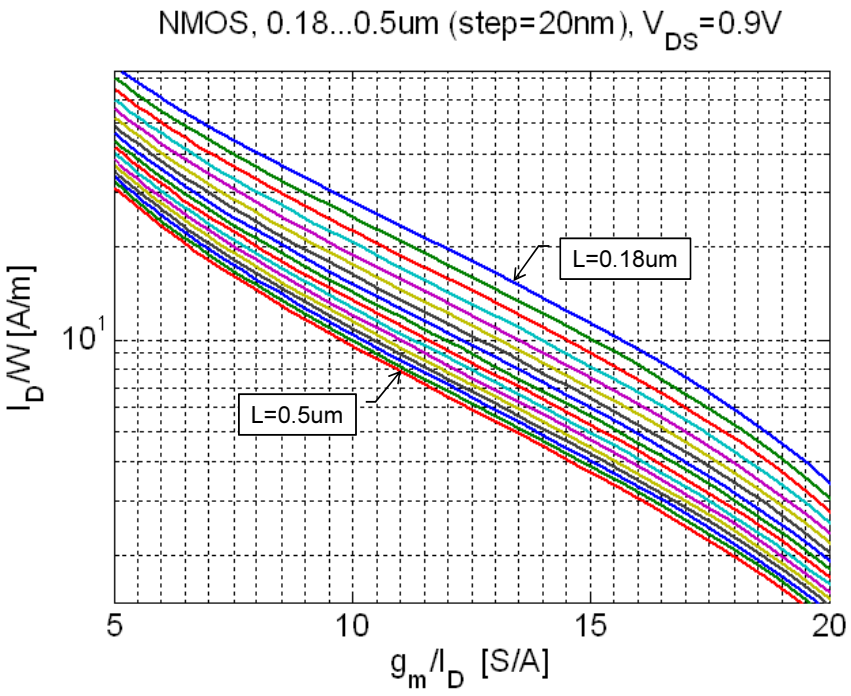
Transit Frequency Chart



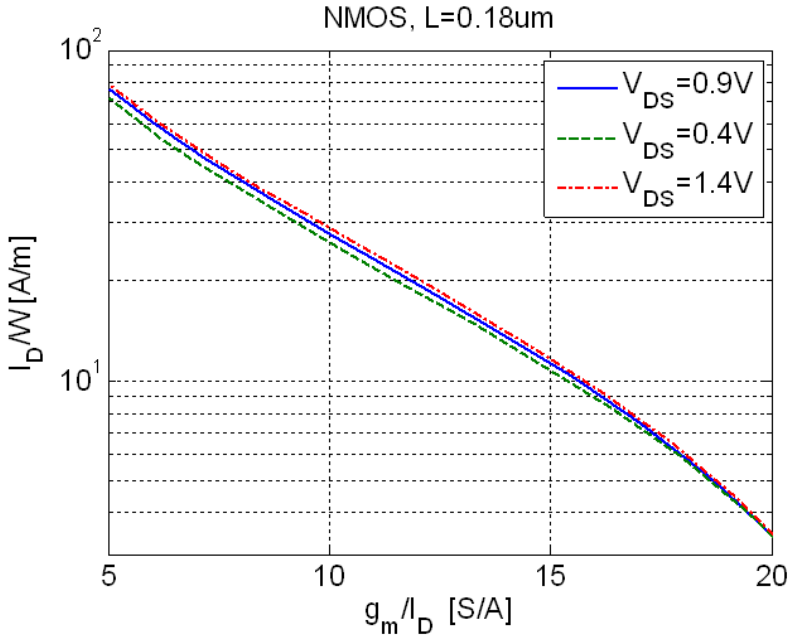
Intrinsic Gain Chart



Current Density Chart

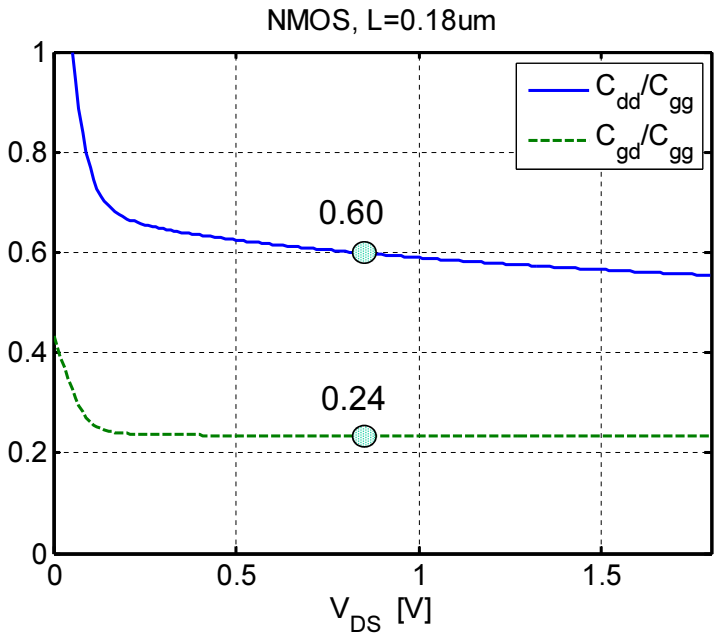


V_{DS} Dependence



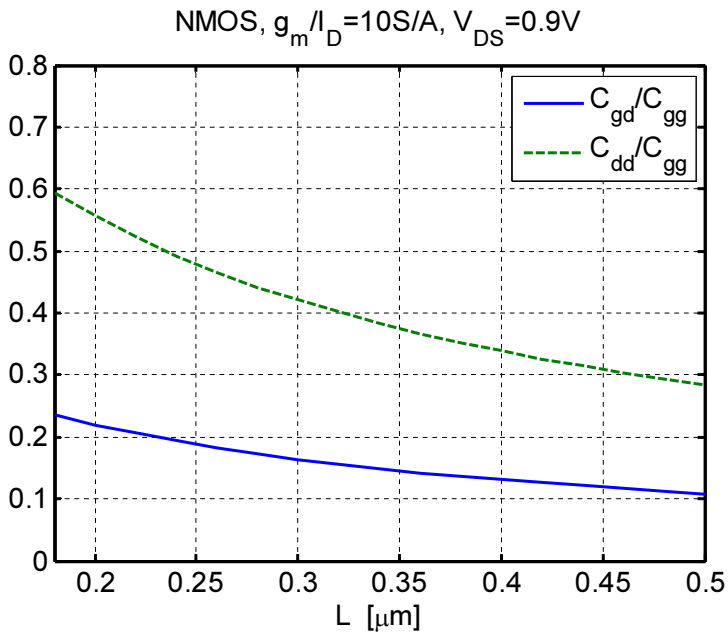
- V_{DS} dependence is relatively weak
- Typically OK to work with data generated for $V_{DD}/2$

Extrinsic Capacitances (1)

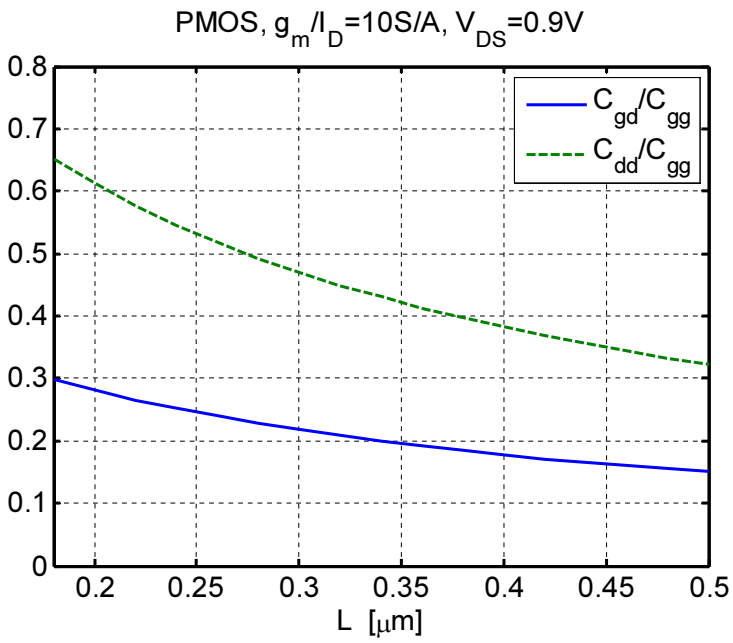


- Again, usually OK to work with estimates taken at $V_{DD}/2$

Extrinsic Capacitances (2)



Extrinsic Capacitances (3)



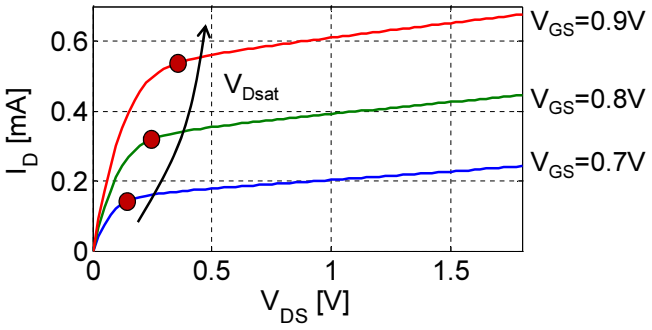
Generic Design Flow

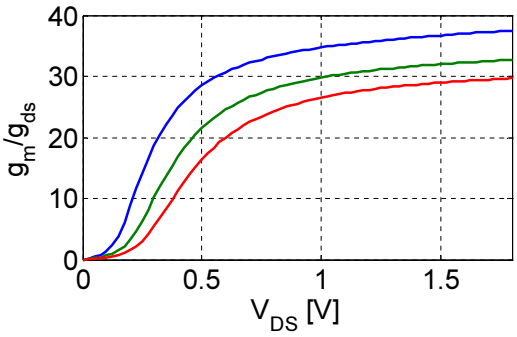
1) Determine g_m (from design objectives)
2) Pick L
   - Short channel → high f_T (high speed)
   - Long channel → high intrinsic gain
3) Pick g_m/I_D (or f_T)
   - Large g_m/I_D → low power, large signal swing (low V_{DSsat})
   - Small g_m/I_D → high f_T (high speed)
4) Determine I_D (from g_m and g_m/I_D)
5) Determine W (from I_D/W)

Many other possibilities exist (depending on circuit specifics, design constraints and objectives)

How about V_{Dsat}?

- V_{Dsat} tells us how much voltage we need across the transistor to operate in saturation
  - "High gain region"
- It is important to note that V_{Dsat} is not crisply defined in modern devices
  - Gradual increase of g_m/g_{ds} with V_{DS}





Relationship Between V_{Dsat} and g_m/I_D

- It turns out that $2/(g_m/I_D)$ is a reasonable first-order estimate for V_{Dsat}

Square Law

$$I_D = K(V_{GS} - V_t)^2$$

$$g_m = 2K(V_{GS} - V_t)$$

$$\frac{2}{(g_m / I_D)} = (V_{GS} - V_t) = V_{DSat}$$

$\therefore$ Consistent with the classical first-order relationship

Weak Inversion

$$I_D = I_{D0} e^{\frac{V_{GS} - V_t}{nV_T}} \left(1 - e^{-\frac{V_{DS}}{V_T}}\right)$$

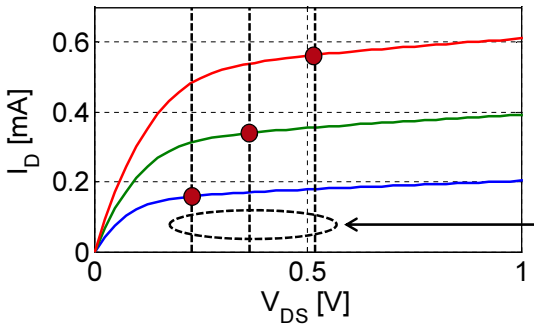
Need about $3V_T$ for saturation

$$g_m = \frac{I_{D0}}{nV_T} e^{\frac{V_{GS} - V_t}{nV_T}} \left(1 - e^{-\frac{V_{DS}}{V_T}}\right)$$

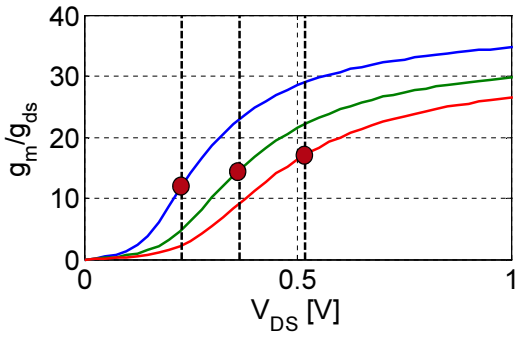
$$\frac{2}{(g_m / I_D)} = 2nV_T \cong 3V_T$$

$\therefore$ Corresponds well with the required minimum V_{DS}

Reality Check

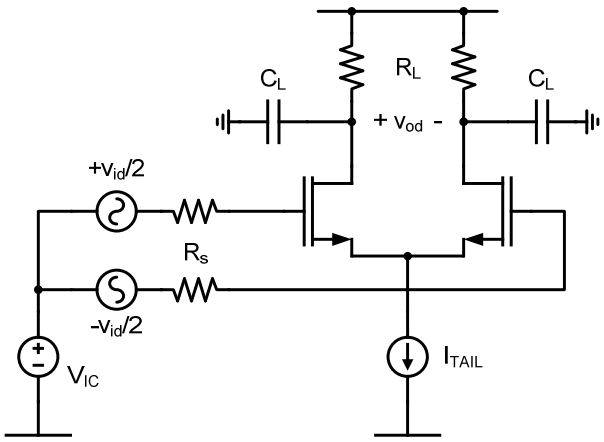


Computed $2/(g_m/I_D)$ values



The SPICE model data confirms that $2/(g_m/I_D)$ is a good estimate for the minimum reasonable V_{DS}

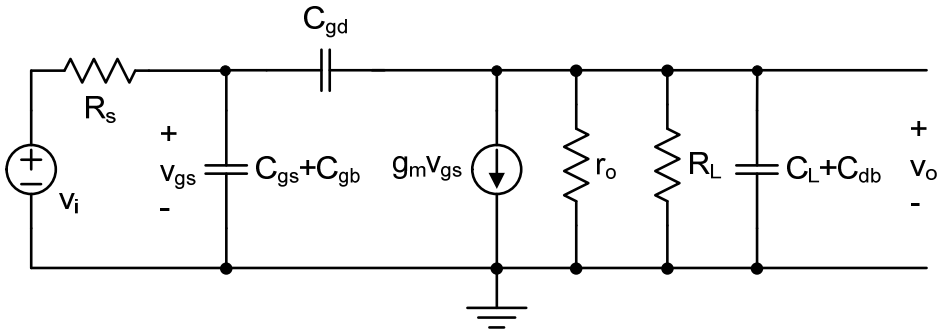
Basic Design Example



Given specifications and objectives

- 0.18μm technology
- Low frequency gain = -4
- R_L=1k, C_L=50fF, R_s=10kΩ
- Maximize bandwidth while keeping $I_{TAIL} \leq 600\mu A$
  - Implies $L=L_{min}=0.18\mu m$
- Determine device width
- Estimate dominant and non-dominant pole

Small-Signal Half-Circuit Model



Calculate g_m and g_m/I_D

$$|A_{v0}| \cong g_m R_L = 4 \quad \Rightarrow \quad g_m = \frac{4}{1k\Omega} = 4mS$$

$$\boxed{\frac{g_m}{I_D} = \frac{4mS}{300\mu A} = 13.3\frac{S}{A}}$$

Why can we Neglect r_o?

$$|A_{v0}| = g_m (R_L \,||\, r_o)$$

$$= g_m \left(\frac{1}{R_L} + \frac{1}{r_o} \right)^{-1}$$

$$\frac{1}{|A_{v0}|} = \frac{1}{g_m R_L} + \frac{1}{g_m r_o}$$

$$\frac{1}{4} = \frac{1}{g_m R_L} + \frac{1}{g_m r_o}$$

- Even at $L=L_{min}= 0.18\mu m$, we have $g_m r_o > 30$
- r_o is negligible in this design problem

Zero and Pole Expressions

High frequency zero (negligible)

$$\omega_z = \frac{g_m}{C_{gd}} >> \omega_T$$

Denominator coefficients

$$b_1 = R_s\left[C_{gs} + C_{gd}\left(1 + |A_{v0}|\right)\right] + R_L(C_L + C_{gd})$$

$$b_2 = R_s R_L(C_{gs}C_L + C_{gs}C_{gd} + C_L C_{gd})$$

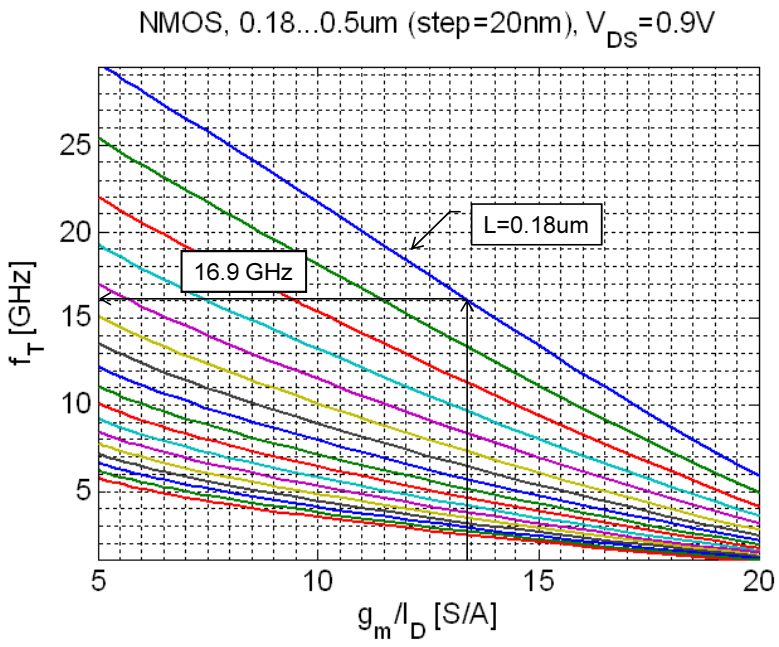
Dominant pole

$$\omega_{p1} \cong \frac{1}{b_1}$$

Nondominant pole

$$\omega_{p2} \cong \frac{b_1}{b_2}$$

Determine C_{gg} via f_T Look-up



Find Capacitances and Plug in

$$C_{gg} = \frac{1}{2\pi}\frac{4mS}{16.9GHz} = 37.8fF$$

$$C_{gd} = \frac{C_{gd}}{C_{gg}} C_{gg} = 0.24 \cdot 37.7fF = 9.0fF$$

$$C_{dd} = \frac{C_{dd}}{C_{gg}} C_{gg} = 0.60 \cdot 37.7fF = 22.6fF$$

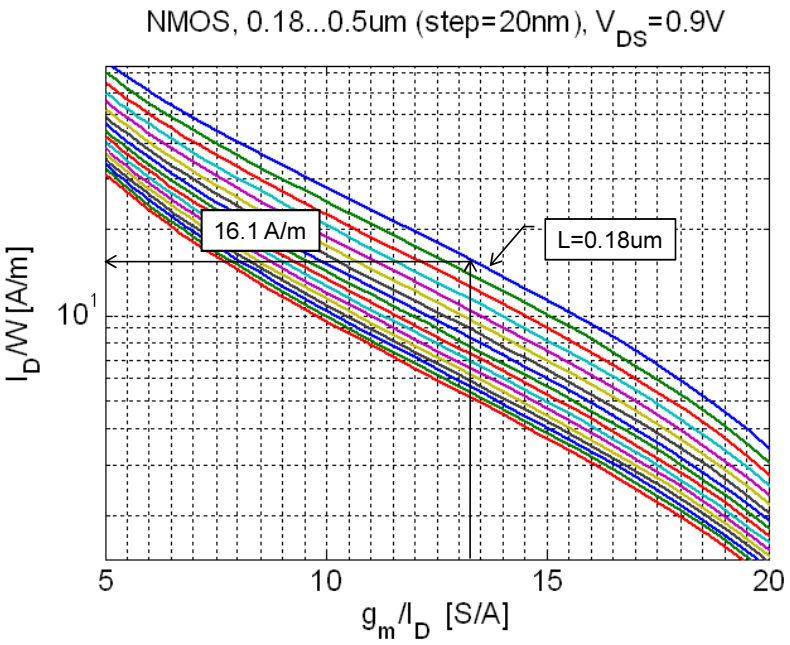
$$C_{db} = C_{dd} - C_{gd} = 13.6fF$$

$$C_{gs} = C_{gg} - C_{gd} = 28.8fF$$

$f_{p1} \cong 200\ MHz$

$f_{p2} \cong 5.8\ GHz$

Device Sizing



A Note on Current Density

- Designing with current density charts in a normalized, width-independent space works because
  - Current density and g_m/I_D are independent of W
    - $I_D/W \sim W/W$
    - $g_m/I_D \sim W/W$
  - There is a one-to-one mapping from g_m/I_D to current density

Square law: $\frac{g_m}{I_D} = \frac{2}{V_{OV}}$ $\qquad \frac{I_D}{W} = \frac{1}{2}\mu C_{ox}\frac{1}{L}V_{OV}^2 = \mu C_{ox}\frac{1}{L}\left(\frac{1}{2}\frac{g_m}{I_D}\right)^{-2}$

General case: $\frac{g_m}{I_D} = f\left(V_{OV}\right)$ $\qquad \frac{I_D}{W} = g\left(V_{OV}\right) = g\left(f^{-1}\left(\frac{g_m}{I_D}\right)\right)$

Matlab Design Script

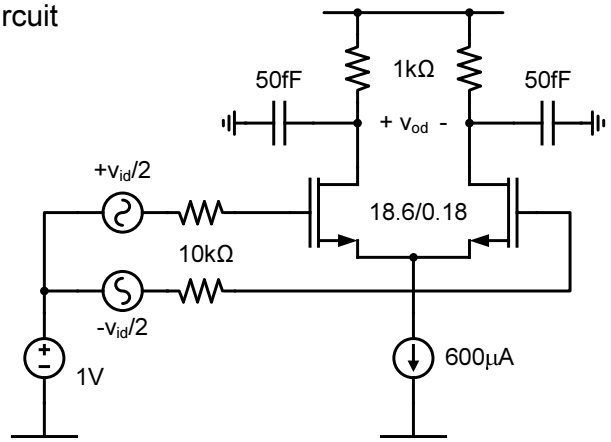
```
% gm/ID design example
clear all; close all;
load 180nch.mat;

% Specs
Av0 = 4; RL = 1e3; CL = 50e-15; Rs = 10e3; ITAIL = 600e-6;

% Component calculations
gm = Av0/RL;
gm_id = gm/(ITAIL/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 calculations
b1 = Rs*(cgs + cgd*(1+Av0))+RL*(CL+cgd);
b2 = Rs*RL*(cgs*CL + cgs*cgd + CL*cgd);
fp1 = 1/2/pi/b1
fp2 = 1/2/pi*b1/b2

% device sizing
id_w = lookup(nch, 'ID_W', 'GM_ID', gm_id);
w = ITAIL/2 / id_w
```

Circuit For Spice Verification

Device width

$$W = \frac{I_D}{\frac{I_D}{W}} = \frac{300\mu A}{16.1A/m} = 18.6\mu m$$

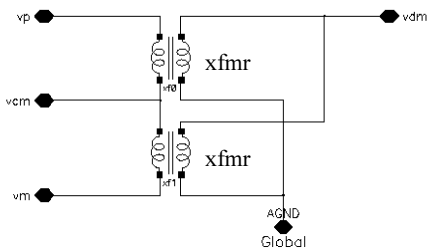
Simulation circuit



Circuit Netlist

```
* gm/id design example

* ee214 device models
.include /usr/class/ee214b/hspice/ee214_hspice.sp

vdd vdd 0 1.8
vic vic 0 1
vid vid 0 ac 1
x1 vid vic vip vim balun
x2 vod voc vop vom balun
rdum vod 0 1gig
it  t 0 600u

m1 vop vgp t 0  nmos214  w=18.6u l=0.18u
m2 vom vgm t 0  nmos214  w=18.6u l=0.18u
rsp vip vgp 10k
rsm vim vgm 10k
rlp vop vdd 1k
rlm vom vdd 1k
clp vop 0 50f
clm vom 0 50f

.op
.ac dec 100 1e6 1000e9

.pz v(vod) vid
.option post brief accurate

.end
```

Ideal Balun



=

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

- Useful for separating CM and DM signal components
- Bi-directional, preserves port impedance
- Uses ideal, inductorless transformers that work down to DC
- Not available in all simulators

Simulated DC Operating Point

element	0:m1	0:m2
model	0:nmos214	0:nmos214
region	Saturati	Saturati
id	300.0000u	300.0000u
vgs	682.4474m	682.4474m
vds	1.1824	1.1824
vbs	-317.5526m	-317.5526m
vth	564.5037m	564.5037m
vdsat	109.0968m	109.0968m
vod	117.9437m	117.9437m
beta	37.2597m	37.2597m
gam eff	583.8490m	583.8490m
gm	4.0718m	4.0718m
gds	100.9678u	100.9678u
gmb	887.2111u	887.2111u
cdtot	20.8290f	20.8290f
cgtot	37.4805f	37.4805f
cstot	42.2382f	42.2382f
cbtot	31.5173f	31.5173f
cgs	26.7862f	26.7862f
cgd	8.9672f	8.9672f

Good agreement!

Design values

g_m = 4 mS

C_{dd} = 22.6 fF
C_{gg} = 37.8 fF
C_{gd} = 9.0 fF

HSpice .OP Capacitance Output Variables

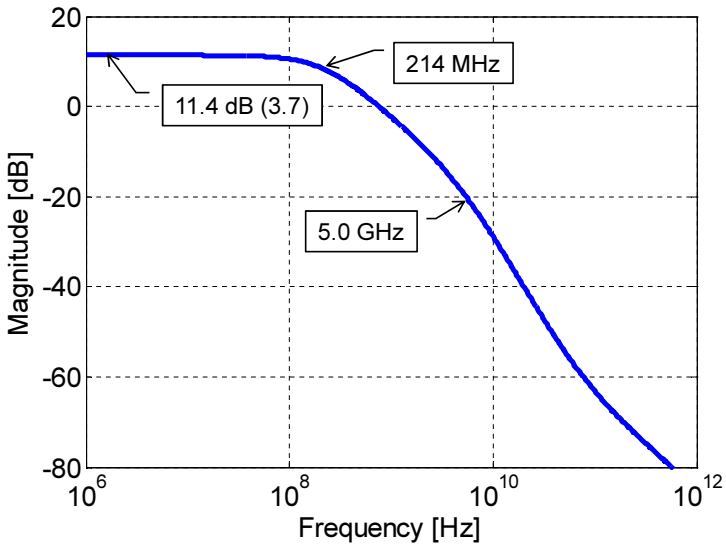
HSpice (.OP)

```
cdtot        20.8290f
cgtot        37.4805f
cstot        42.2382f
cbtot        31.5173f
cgs          26.7862f
cgd           8.9672f
```

Corresponding Small Signal Model Elements

cdtot $\equiv C_{gd} + C_{db}$
cgtot $\equiv C_{gs} + C_{gd} + C_{gb}$
cstot $\equiv C_{gs} + C_{sb}$
cbtot $\equiv C_{gb} + C_{sb} + C_{db}$
cgs $\equiv C_{gs}$
cgd $\equiv C_{gd}$

Simulated AC Response



- Calculated values: $|A_{v0}|$=12 dB (4.0), f_{p1} = 200 MHz, f_{p2}= 5.8 GHz

Plotting HSpice Results in Matlab

```
clear all;
close all;
addpath('/usr/class/ee214b/matlab/hspice_toolbox');

h = loadsig('gm_id_example1.ac0');
lssig(h)

f = evalsig(h,'HERTZ');
vod = evalsig(h,'vod');
magdb = 20*log10(abs(vod));
av0 = abs(vod(1))
f3dB = interp1(magdb, f, magdb(1)-3, 'spline')

figure(1);
semilogx(f, magdb, 'linewidth', 3);
xlabel('Frequency [Hz]');
ylabel('Magnitude [dB]');
axis([1e6 1e12 -80 20]);
grid;
```


Using .pz Analysis

Netlist statement

```
.pz v(vod) vid
```

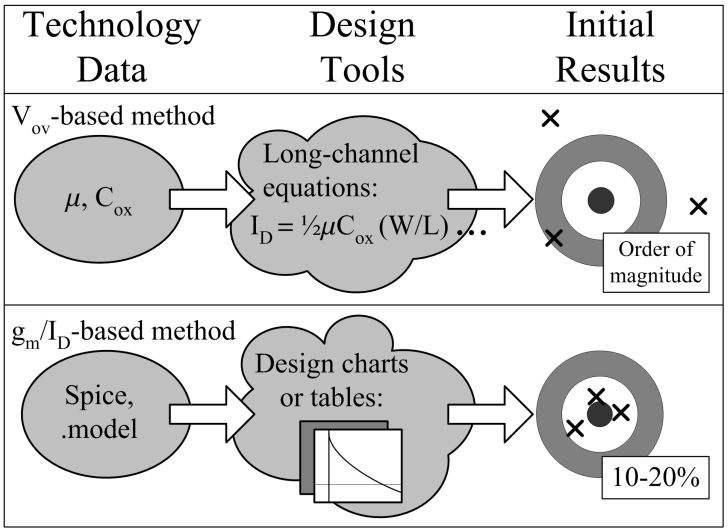
Output

```
**************************************************
 input =  0:vid          output = v(vod)

      poles (rad/sec)                poles ( hertz)
 real            imag            real            imag
 -1.35289g       0.              -215.319x       0.
 -31.6307g       0.              -5.03418g       0.

      zeros (rad/sec)                zeros ( hertz)
 real            imag            real            imag
 445.734g        0.              70.9407g        0
```

Observations

- The design is essentially right on target!
  - Typical discrepancies are no more than 10-20%, due to V_{DS} dependencies, finite output resistance, etc.
- We accomplished this by using pre-computed spice data in the design process
- Even if discrepancies are more significant, there's always the possibility to track down the root causes
  - Hand calculations are based on parameters that also exist in Spice, e.g. g_m/I_D, f_T, etc.
  - Different from square law calculations using μC_{ox}, V_{OV}, etc.
    - Based on artificial parameters that do not exist or have no significance in the spice model

Comparison



References

- F. Silveira et al. "A g_m/I_D based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA," IEEE J. Solid-State Circuits, Sep. 1996, pp. 1314-1319.
- D. Foty, M. Bucher, D. Binkley, "Re-interpreting the MOS transistor via the inversion coefficient and the continuum of g_{ms}/I_d," Proc. Int. Conf. on Electronics, Circuits and Systems, pp. 1179-1182, Sep. 2002.
- B. E. Boser, "Analog Circuit Design with Submicron Transistors," IEEE SSCS Meeting, Santa Clara Valley, May 19, 2005, http://www.ewh.ieee.org/r6/scv/ssc/May1905.htm
- P. Jespers, The g_m/I_D Methodology, a sizing tool for low-voltage analog CMOS Circuit*s*, Springer, 2010.
- T. Konishi, K. Inazu, J.G. Lee, M. Natsu, S. Masui, and B. Murmann, "Optimization of High-Speed and Low-Power Operational Transconductance Amplifier Using g_m/I_D Lookup Table Methodology," IEICE Trans. Electronics, Vol. E94-C, No.3, Mar. 2011.