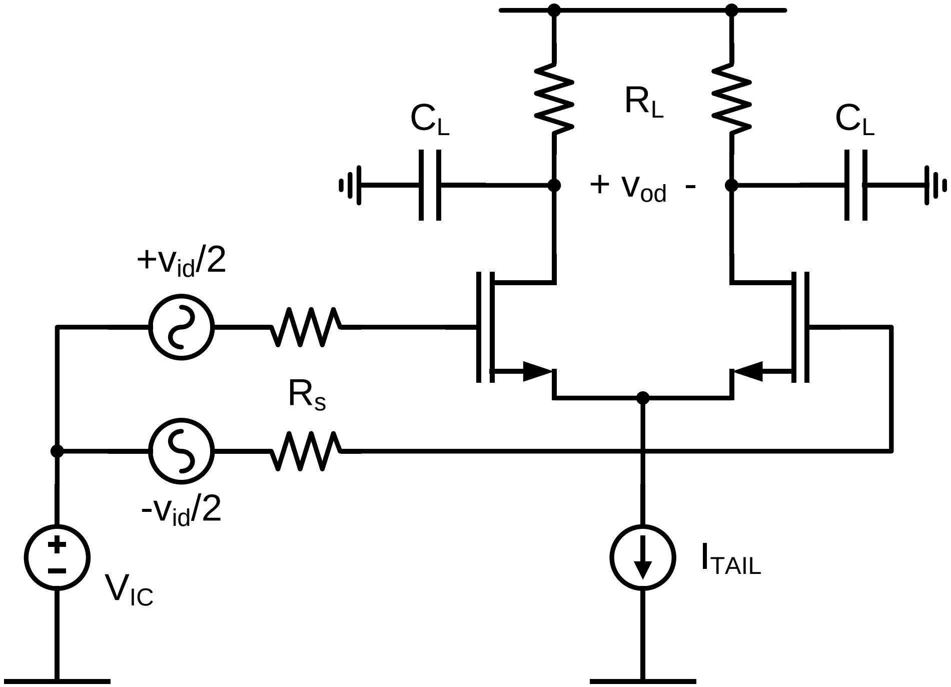
Introduction to G_m/I_D-based sizing

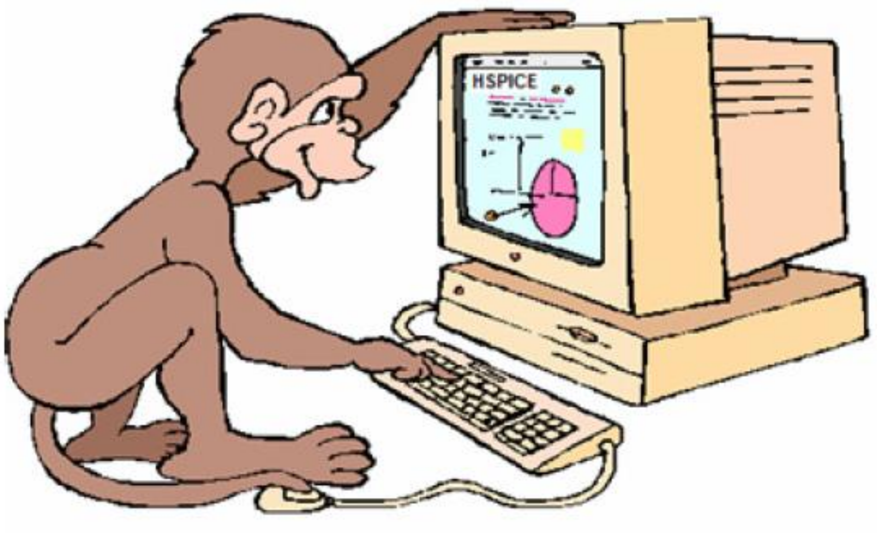
Motivation: Basic Design Example



- Given specifications and objectives
  - 0.18μm technology
  - Low frequency gain = -4
  - R_L=1k, C_L=50fF, R_s=10kΩ
  - Maximize bandwidth while keeping $I_{TAIL} \leq 600\mu A$
  - Determine W/L
  - Estimate dominant and non-dominant pole

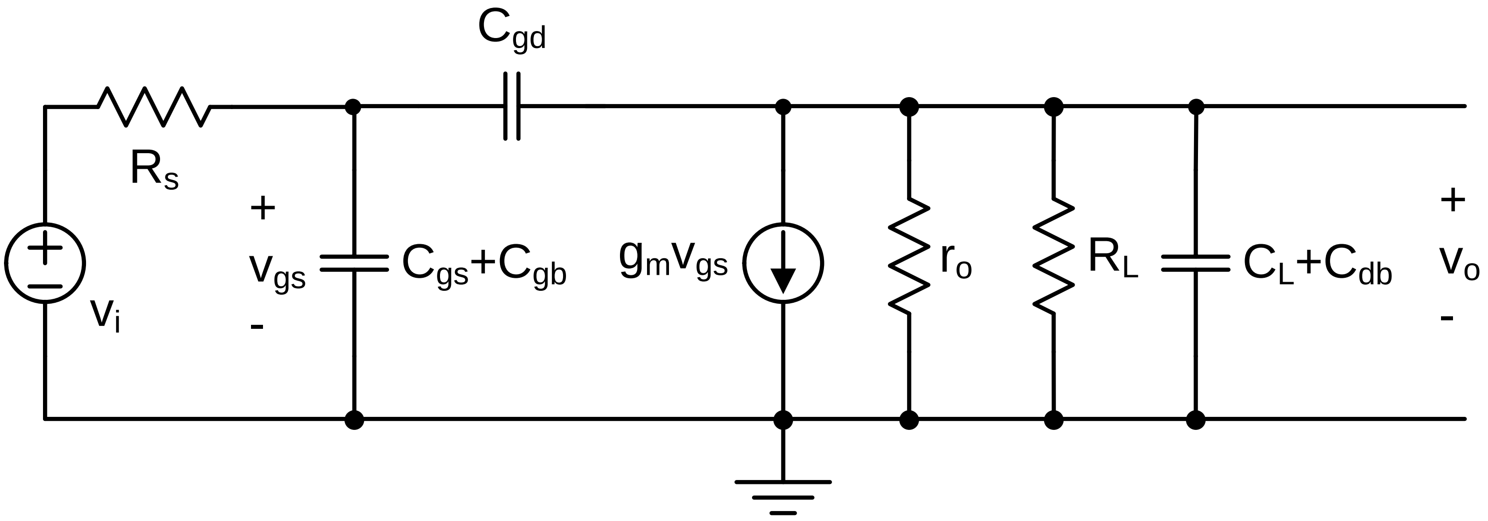
To Be Avoided: Spice Monkeying

- One way to solve this problem is to “poke around” in Spice and play this out like a video game…
- Issues
  - Learn nothing about fundamental tradeoffs and optimality
  - Will not detect simulation or modeling errors



Better: Systematic Design

- Start with a circuit model
- Establish links between design specs and model parameters
- Establish links between model parameters and transistor parameters
  - This is where things get problematic…



Textbook Transistor Model

$$\boxed{I_D = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{GS}-V_t)^2(1+\lambda V_{DS})}$$

$$g_m = \frac{dI_D}{dV_{GS}} = \mu C_{ox}\frac{W}{L}V_{OV}\left(1+\lambda V_{DS}\right) = \frac{2I_D}{V_{OV}} \qquad V_{OV} = V_{GS} - V_t$$

$$g_o = \frac{dI_D}{dV_{DS}} = \frac{1}{2}\mu C_{ox}\frac{W}{L}V_{OV}{}^2 \cdot \lambda = \frac{\lambda I_D}{1+\lambda V_{DS}} \cong \lambda I_D$$

$$C_{gs} = \frac{2}{3}WLC_{ox}$$

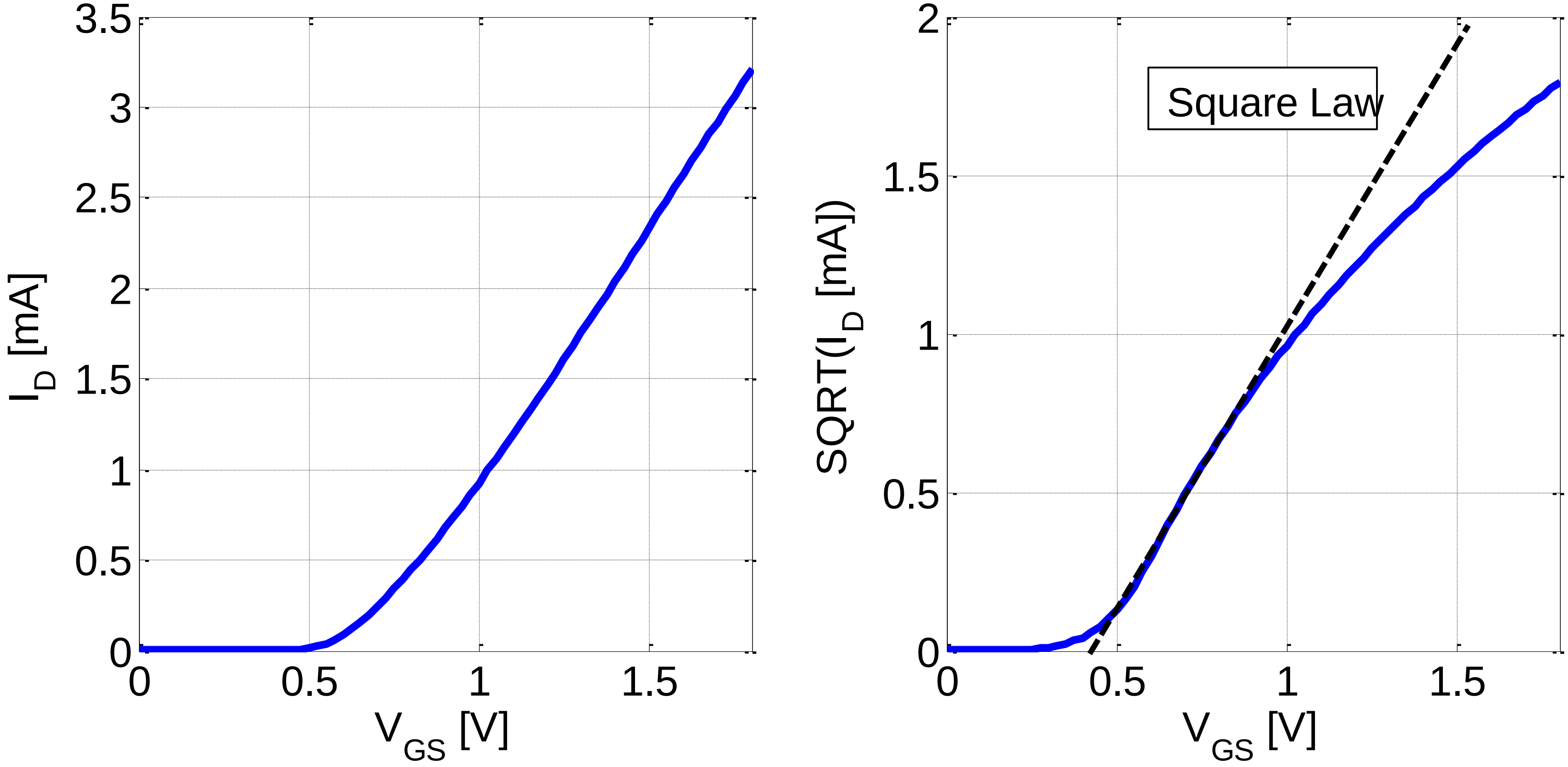
etc.

What are μC_{ox} (KP) and λ (LAMBDA)?

```
.MODEL nmos214 nmos
+acm     = 3               hdif    = 0.32e-6         LEVEL   = 49
+VERSION = 3.1             TNOM    = 27              TOX     = 4.1E-9
+XJ      = 1E-7            NCH     = 2.3549E17       VTH0    = 0.3618397
+K1      = 0.5916053       K2      = 3.225139E-3     K3      = 1E-3
+K3B     = 2.3938862       W0      = 1E-7            NLX     = 1.776268E-7
+DVT0W   = 0               DVT1W   = 0               DVT2W   = 0
+DVT0    = 1.3127368       DVT1    = 0.3876801       DVT2    = 0.0238708
+U0      = 256.74093       UA      = -1.585658E-9    UB      = 2.528203E-18
+UC      = 5.182125E-11    VSAT    = 1.003268E5      A0      = 1.981392
+AGS     = 0.4347252       B0      = 4.989266E-7     B1      = 5E-6
+KETA    = -9.888408E-3    A1      = 6.164533E-4     A2      = 0.9388917
+RDSW    = 128.705483      PRWG    = 0.5             PRWB    = -0.2
+WR      = 1               WINT    = 0               LINT    = 1.617316E-8
+XL      = 0               XW      = -1E-8           DWG     = -5.383413E-9
+DWB     = 9.111767E-9     VOFF    = -0.0854824      NFACTOR = 2.2420572
+CIT     = 0               CDSC    = 2.4E-4          CDSCD   = 0
+CDSCB   = 0               ETA0    = 2.981159E-3     ETAB    = 9.289544E-6
+DSUB    = 0.0159753       PCLM    = 0.7245546       PDIBLC1 = 0.1568183
+PDIBLC2 = 2.543351E-3     PDIBLCB = -0.1            DROUT   = 0.7445011
+PSCBE1  = 8E10            PSCBE2  = 1.876443E-9     PVAG    = 7.200284E-3
+DELTA   = 0.01            RSH     = 6.6             MOBMOD  = 1
+PRT     = 0               UTE     = -1.5            KT1     = -0.11
+KT1L    = 0               KT2     = 0.022           UA1     = 4.31E-9
+UB1     = -7.61E-18       UC1     = -5.6E-11        AT      = 3.3E4
+WL      = 0               WLN     = 1               WW      = 0
+WWN     = 1               WWL     = 0               LL      = 0
+LLN     = 1               LW      = 0               LWN     = 1
+LWL     = 0               CAPMOD  = 2               XPART   = 1
+CGDO    = 4.91E-10        CGSO    = 4.91E-10        CGBO    = 1E-12
+CJ      = 9.652028E-4     PB      = 0.8             MJ      = 0.3836899
+CJSW    = 2.326465E-10    PBSW    = 0.8             MJSW    = 0.1253131
+CJSWG   = 3.3E-10         PBSWG   = 0.8             MJSWG   = 0.1253131
+CF      = 0               PVTH0   = -7.714081E-4    PRDSW   = -2.5827257
+PK2     = 9.619963E-4     WKETA   = -1.060423E-4    LKETA   = -5.373522E-3
+PU0     = 4.5760891       PUA     = 1.469028E-14    PUB     = 1.783193E-23
+PVSAT   = 1.19774E3       PETA0   = 9.968409E-5     PKETA   = -2.51194E-3
+nlev    = 3               kf      = 0.5e-25
```

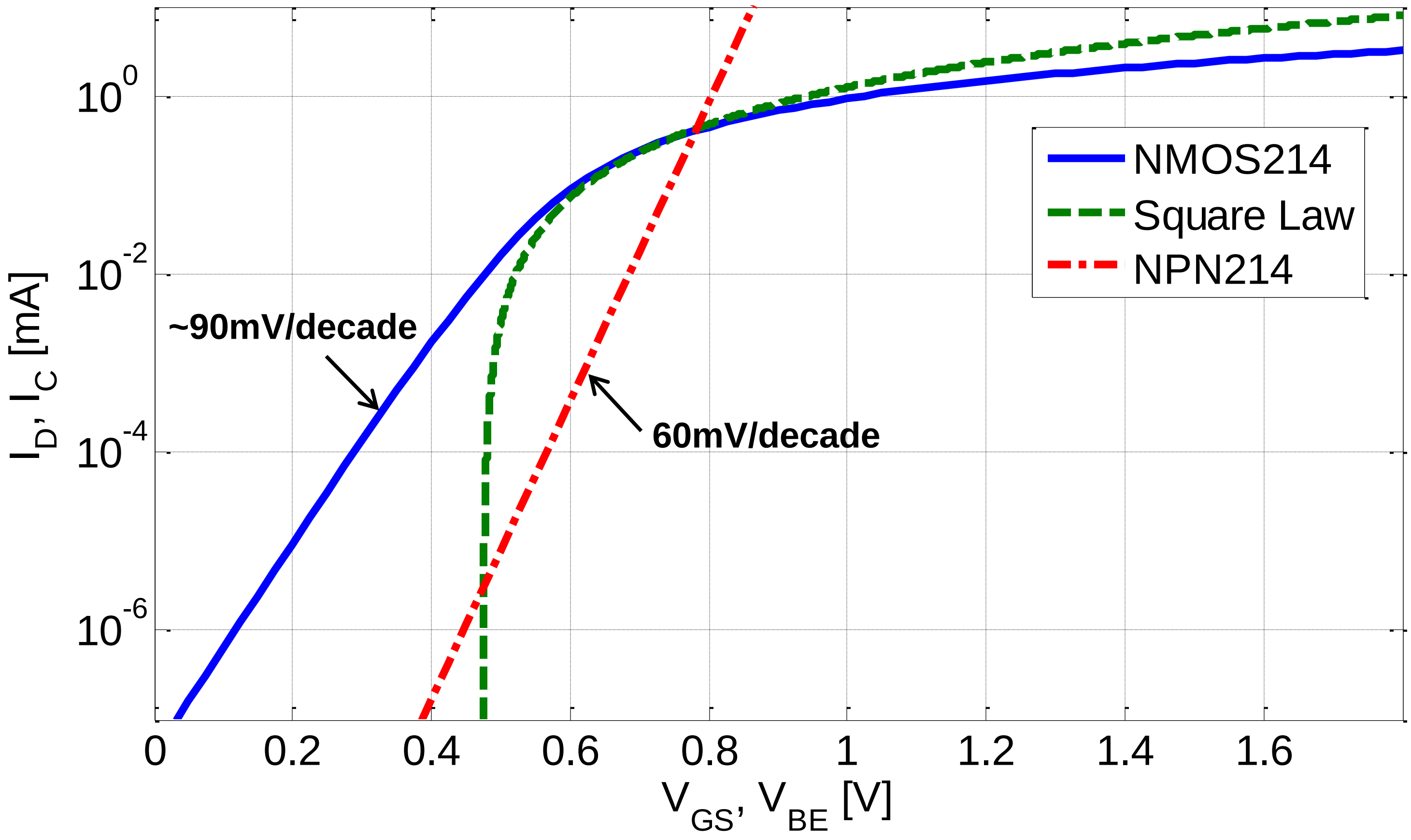
- Even for a relatively old 0.18μm process, the models used in simulation are quite complex
- The model card shown on the left is a 110-parameter BSIM3v3 model
  - More recent models require even more parameters (e.g. PSP, BSIM6)
  - KP and LAMBDA are nowhere to be found
- It turns out that the I-V characteristics of a modern MOSFET cannot be accurately described by the square law

Simulation (NMOS, 5/0.18μm, V_{DS}=1.8V)

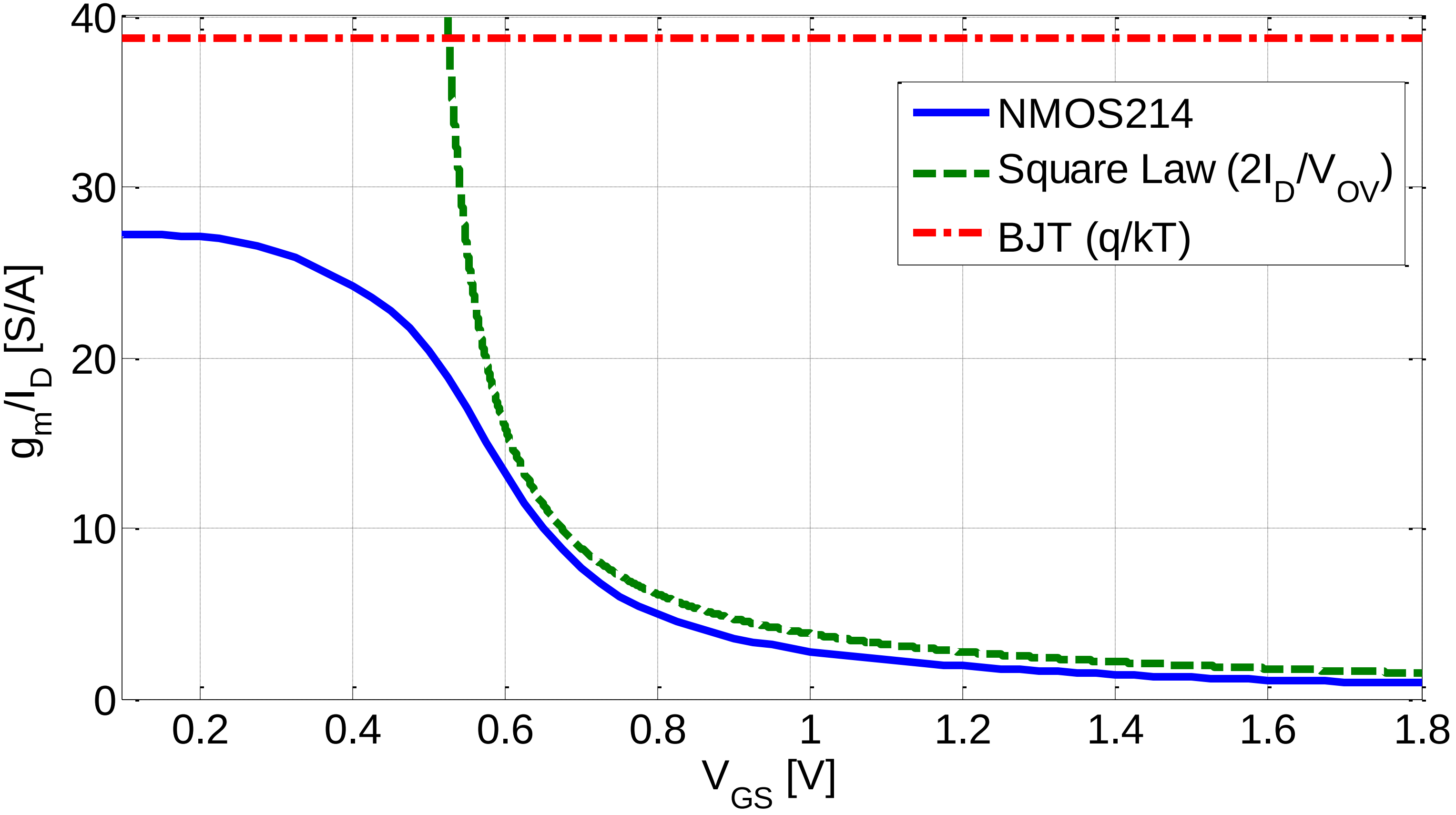


- Two observations
  - The transistor does not abruptly turn off at some V_t
  - The current is not perfectly quadratic in (V_{GS}–V_t)

Current on a Log Scale

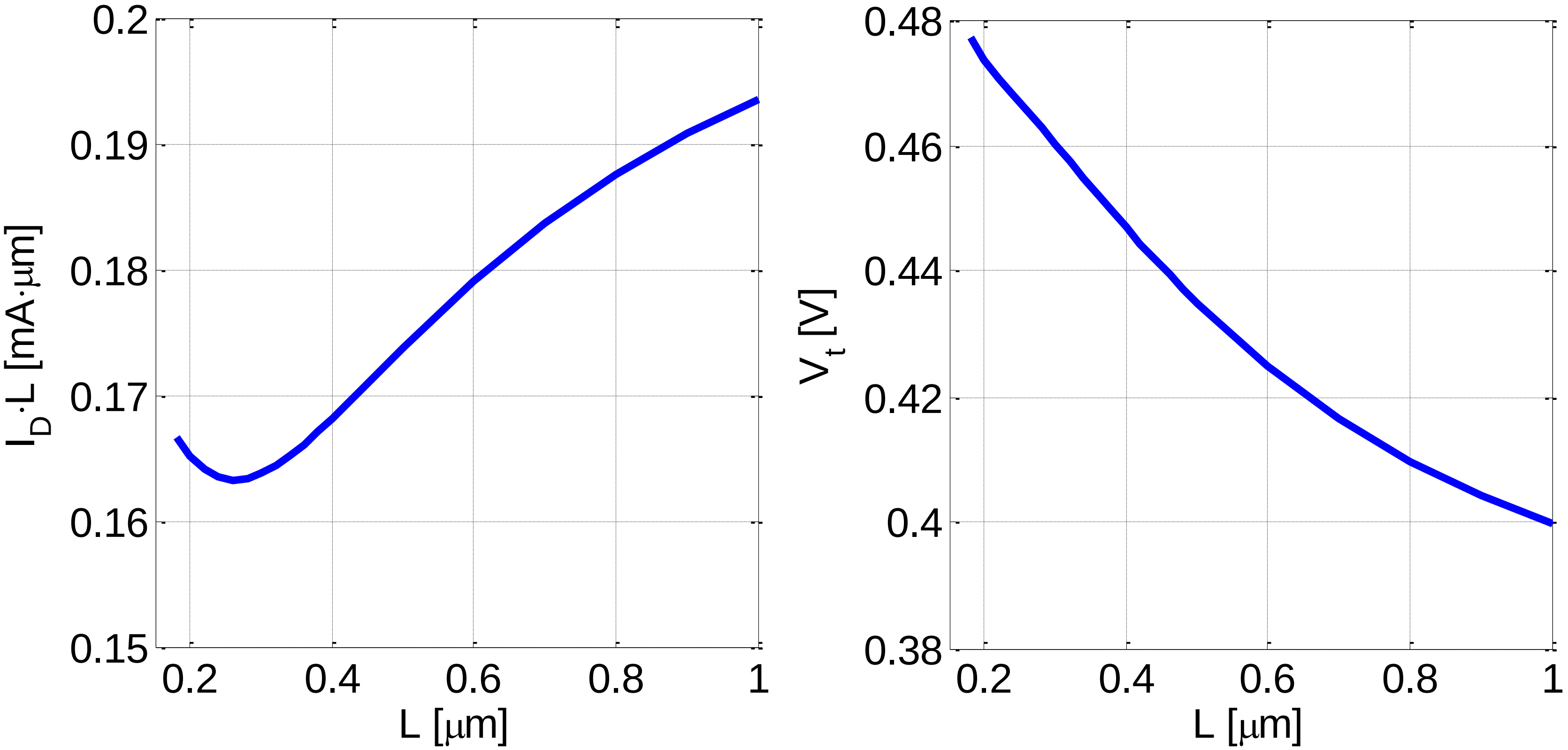


g_m/I_D



- The square law fails miserably at predicting g_m/I_D for low V_{GS}

Length Scaling and V_t

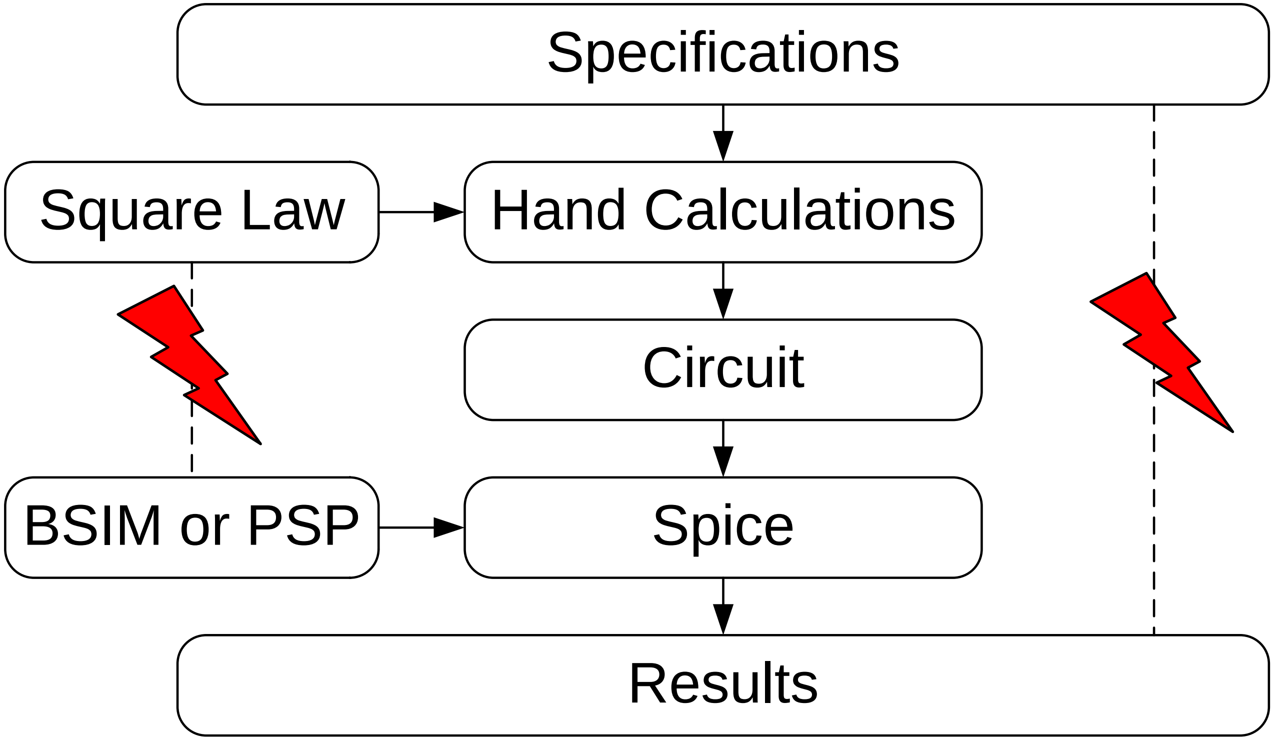


- The current does not scale perfectly with 1/L ($I_D \cdot L \neq$ const.)
- The threshold voltage of the device depends on the channel length

Cause of these Complications

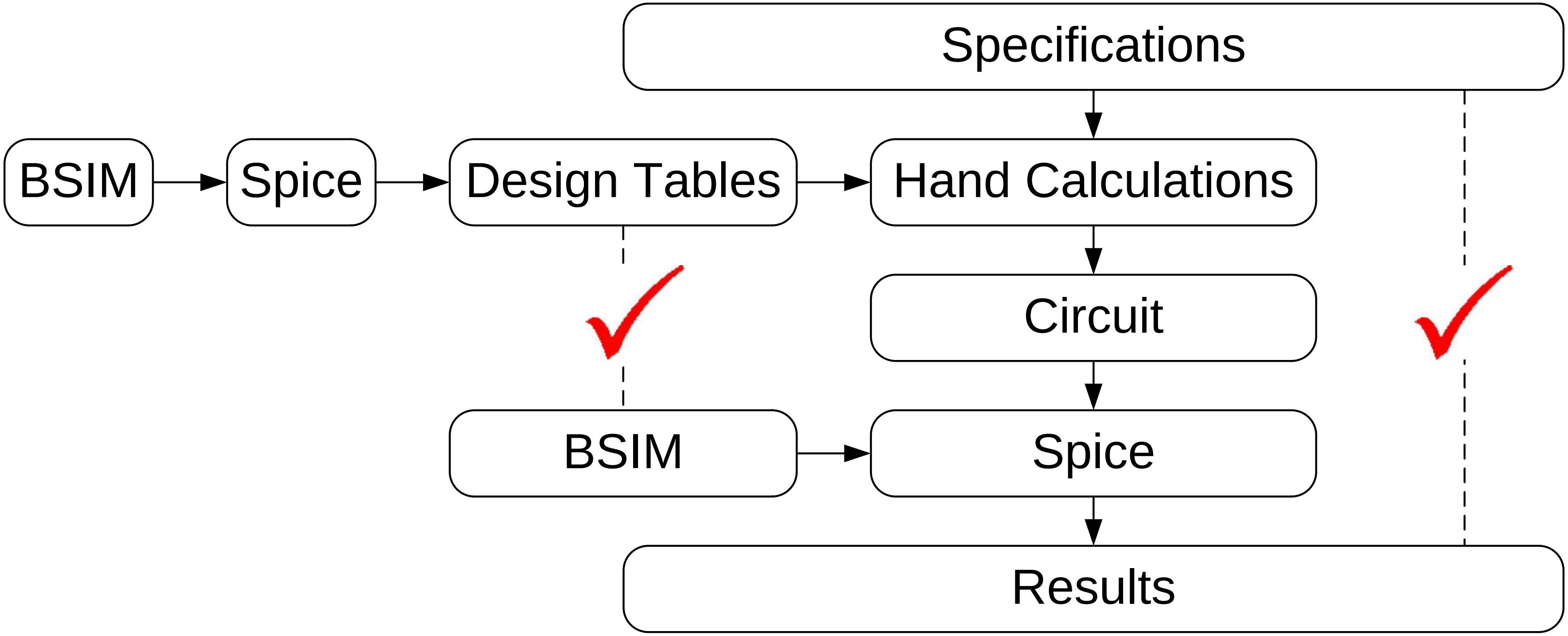
- Weak inversion
  - For V_{GS} below of near V_t, the current is caused by diffusion instead of drift; MOSFET behaves similar to BJT
- Moderate inversion
  - A wild mix of drift and diffusion currents
- Strong inversion
  - Mostly drift current, but short channel effects complicate the transistor behavior
    - Velocity saturation due to high lateral field
    - Mobility degradation due to high vertical field
- In addition, there are several effects related to small geometries
  - Strong V_{DS} dependence of I_D, r_o and V_t (DIBL)
  - V_t depends on channel length (SCE and RSCE)

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

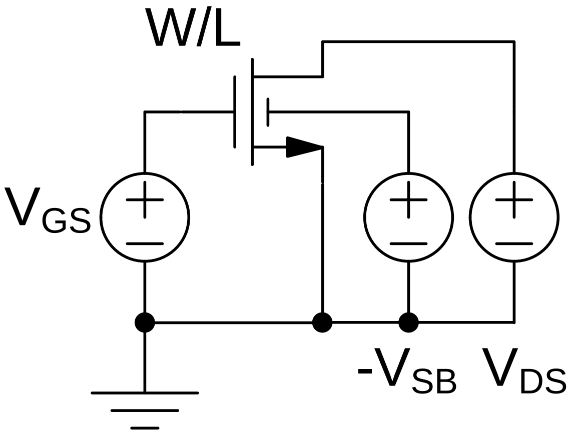
The Solution



- 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('-cbsbo(mn)-cgsbo(mn)')
```



Store Data in a Matlab Structure

```
>> load 180n.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]
     VSB: [11x1 double]
       L: [22x1 double]
       W: 5

>> size(nch.ID)

ans =
    22    73    73    11
```

Four-dimensional arrays

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

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

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

…

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, VSB)
    (2) Lookup of arbitrary ratios of parameters, e.g. GM_ID, GM_CGG at given
        (L, VGS, VDS, VSB)
    (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, VSB) 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)
    VSB = 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 that I promote
  - 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 $\mathbf{g_m/I_D}$ as a proxy

Figures of Merit for Design

Square Law

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

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

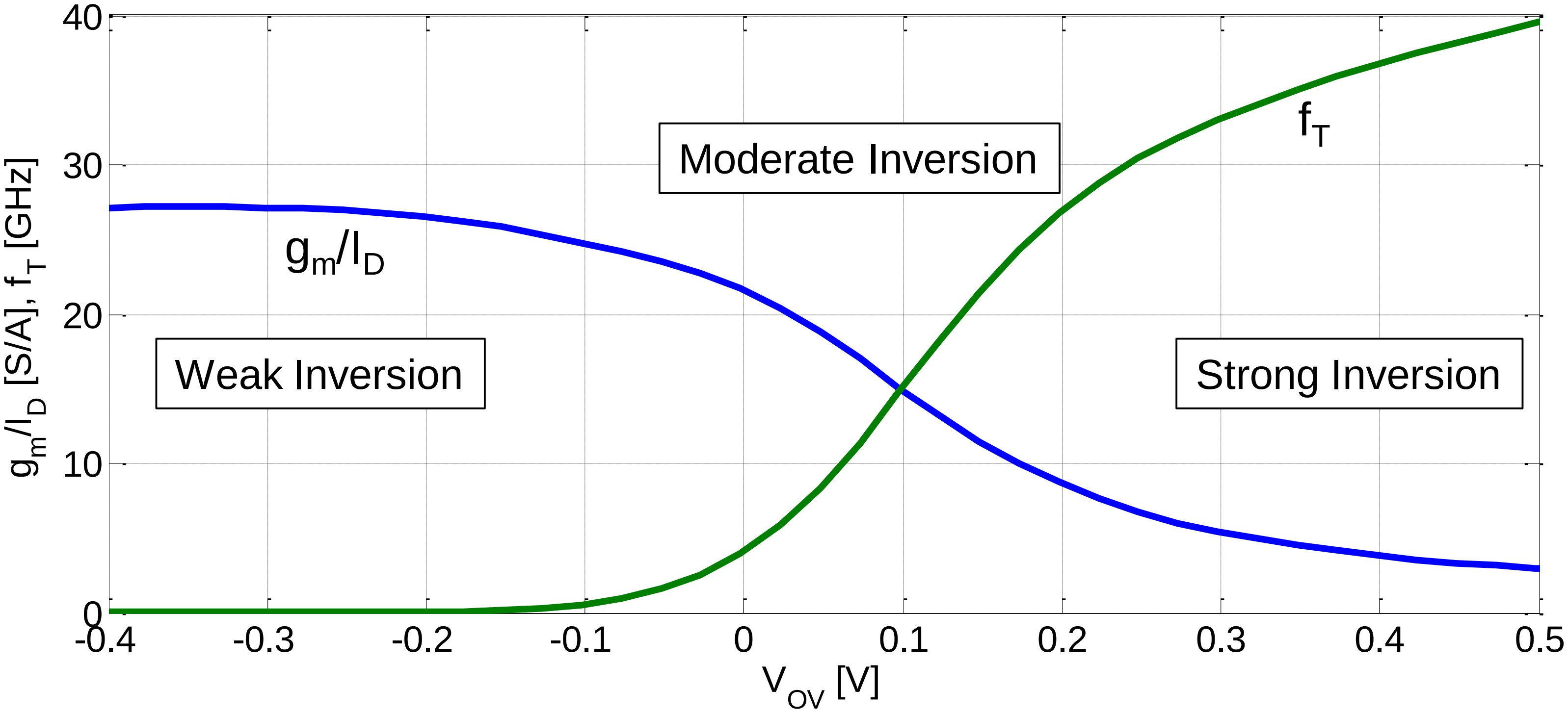
- Transit frequency
  - Want large g_m, without large C_{gg}

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

- Intrinsic gain
  - Want large g_m, but no g_o

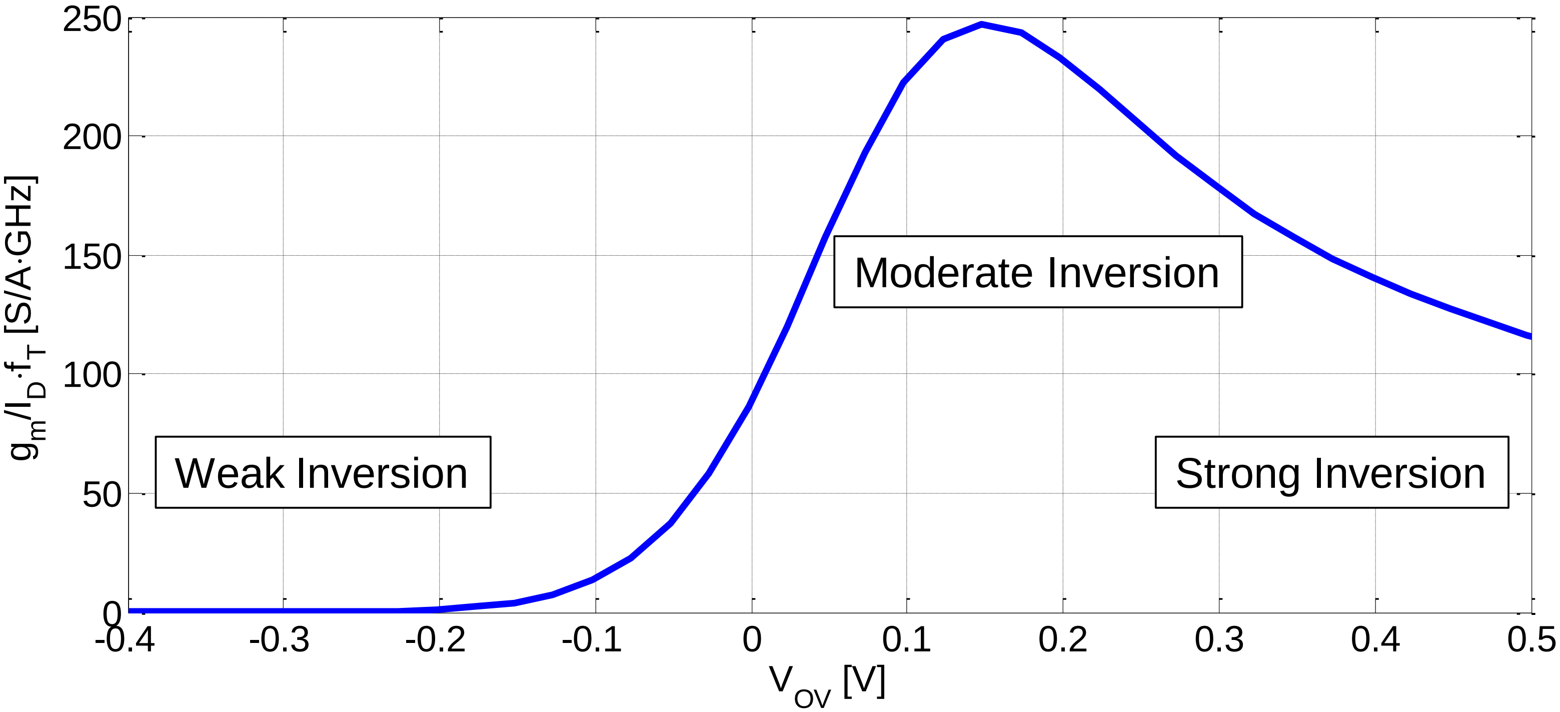
$$\boxed{\frac{g_m}{g_o}} \qquad \cong \frac{2}{\lambda V_{OV}}$$

Design Tradeoff: g_m/I_D and f_T



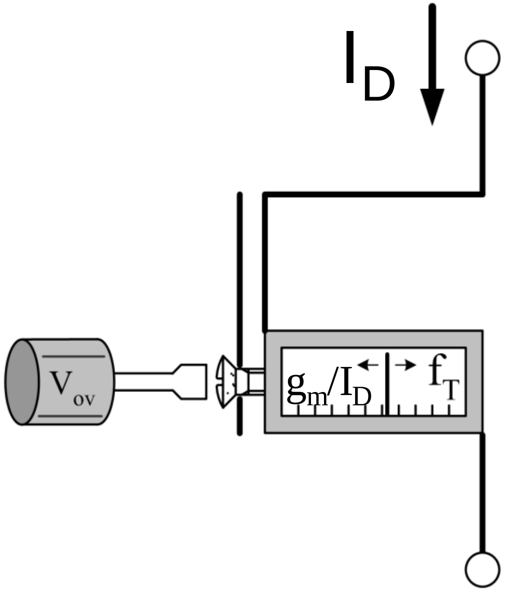
- Weak inversion: Large g_m/I_D (>20 S/A), but small f_T
- Strong inversion: Small g_m/I_D (<10 S/A), but large f_T

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)

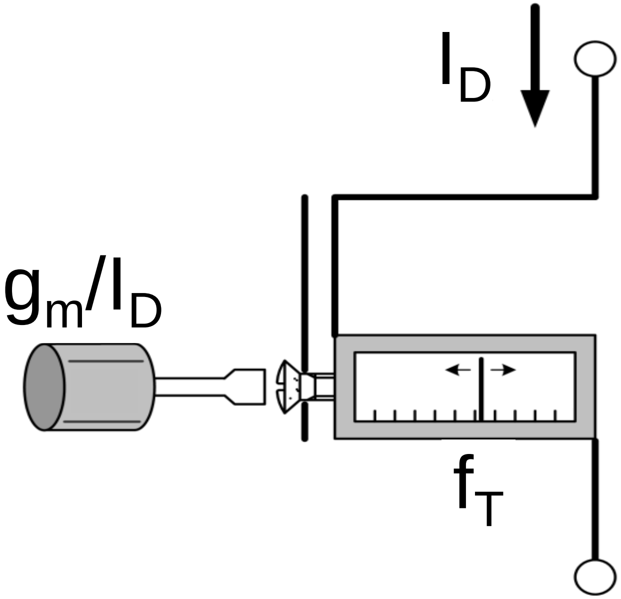
Design in a Nutshell

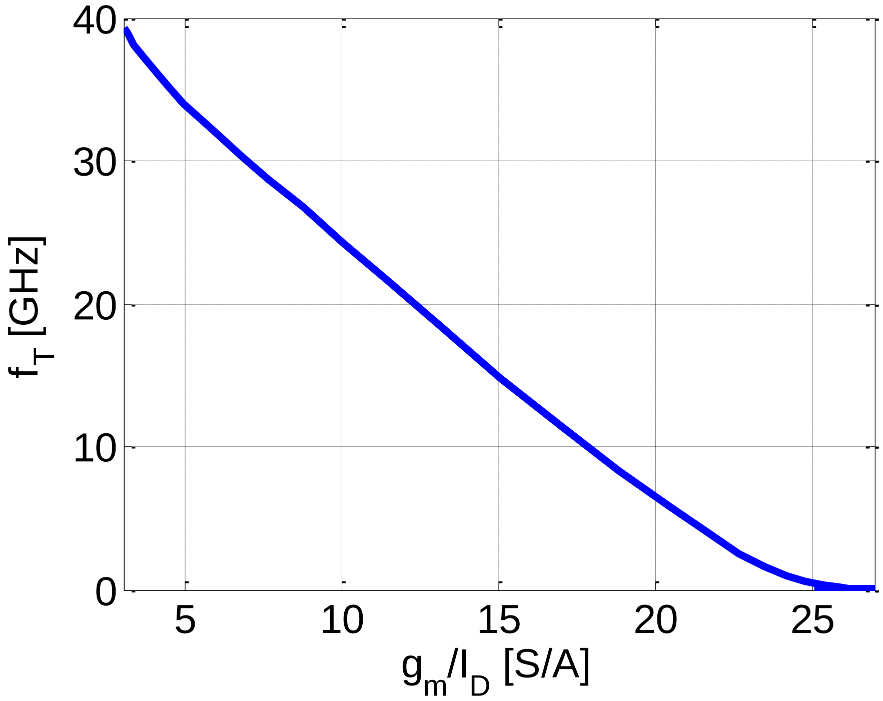


- Choose the inversion level according to the proper tradeoff between speed (f_T) and efficiency (g_m/I_D) for the given circuit
- The inversion level is fully determined by the gate overdrive V_{OV}
  - But, V_{OV} is not a very interesting parameter outside the square law framework; not much can be 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}
  - Even V_{Dsat} can be estimated using $2 / (g_m/I_D)$

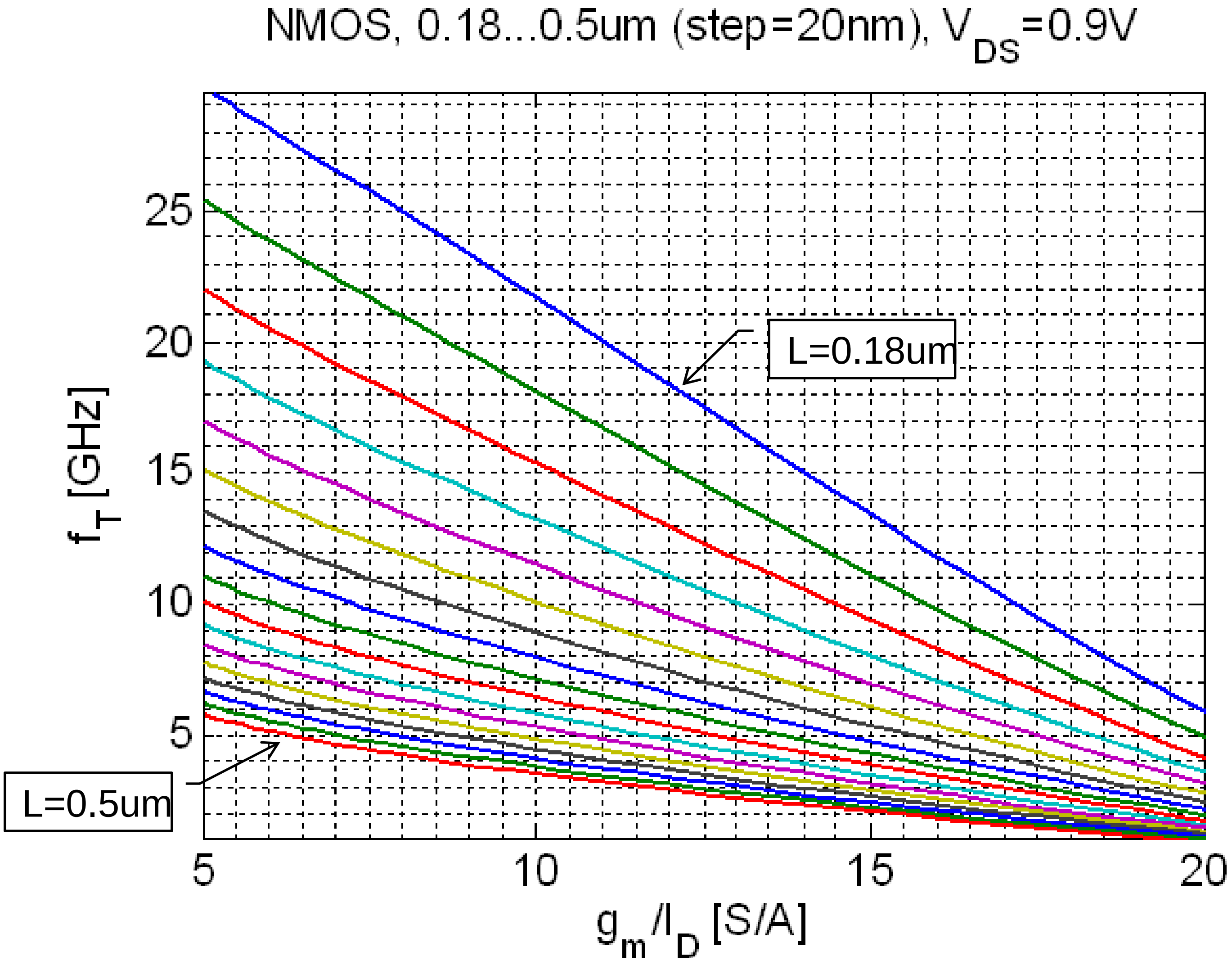




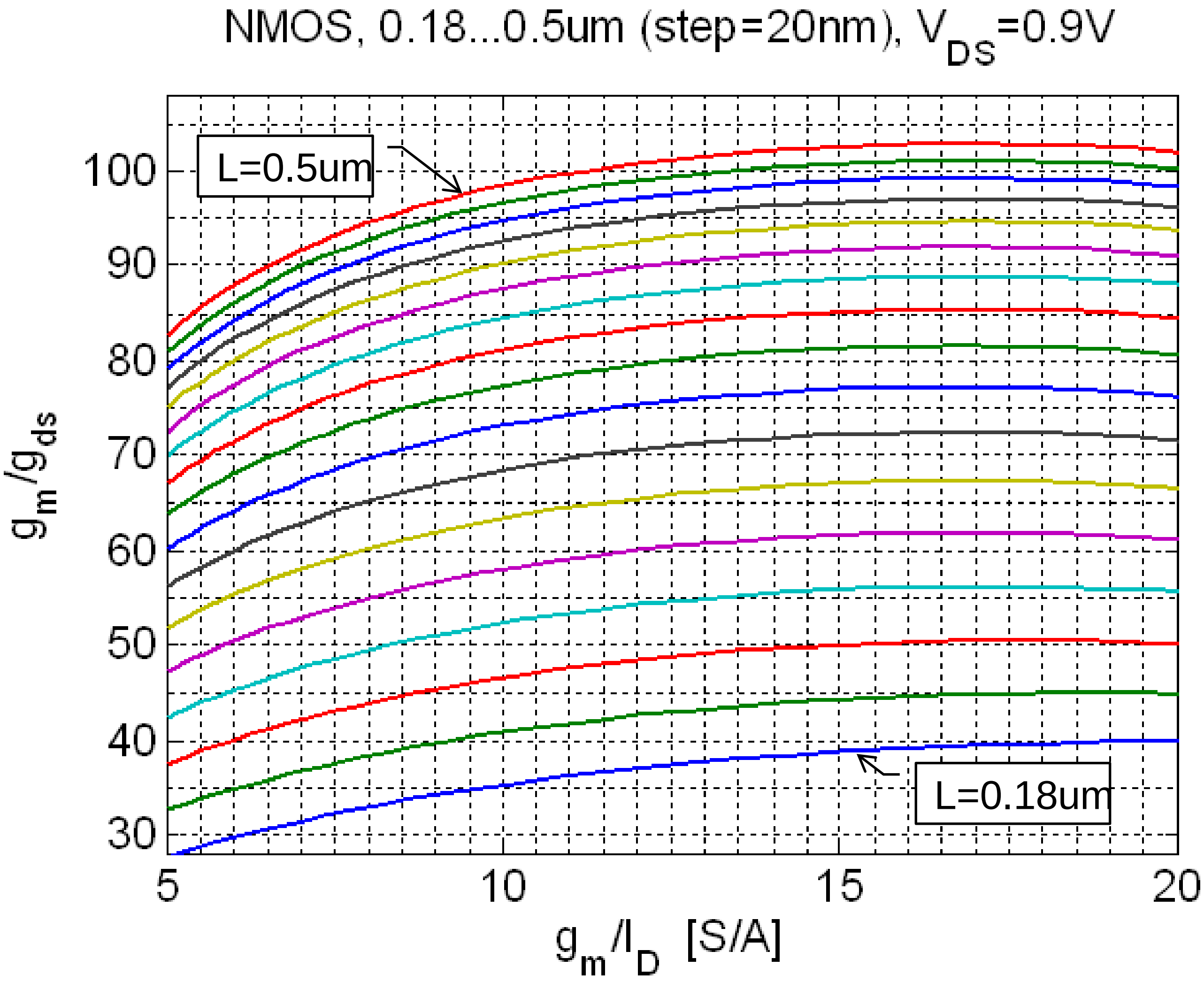
g_m/I_D-centric Technology Characterization

- Plot 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 plot 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 plot that links g_m/I_D and current density I_D/W

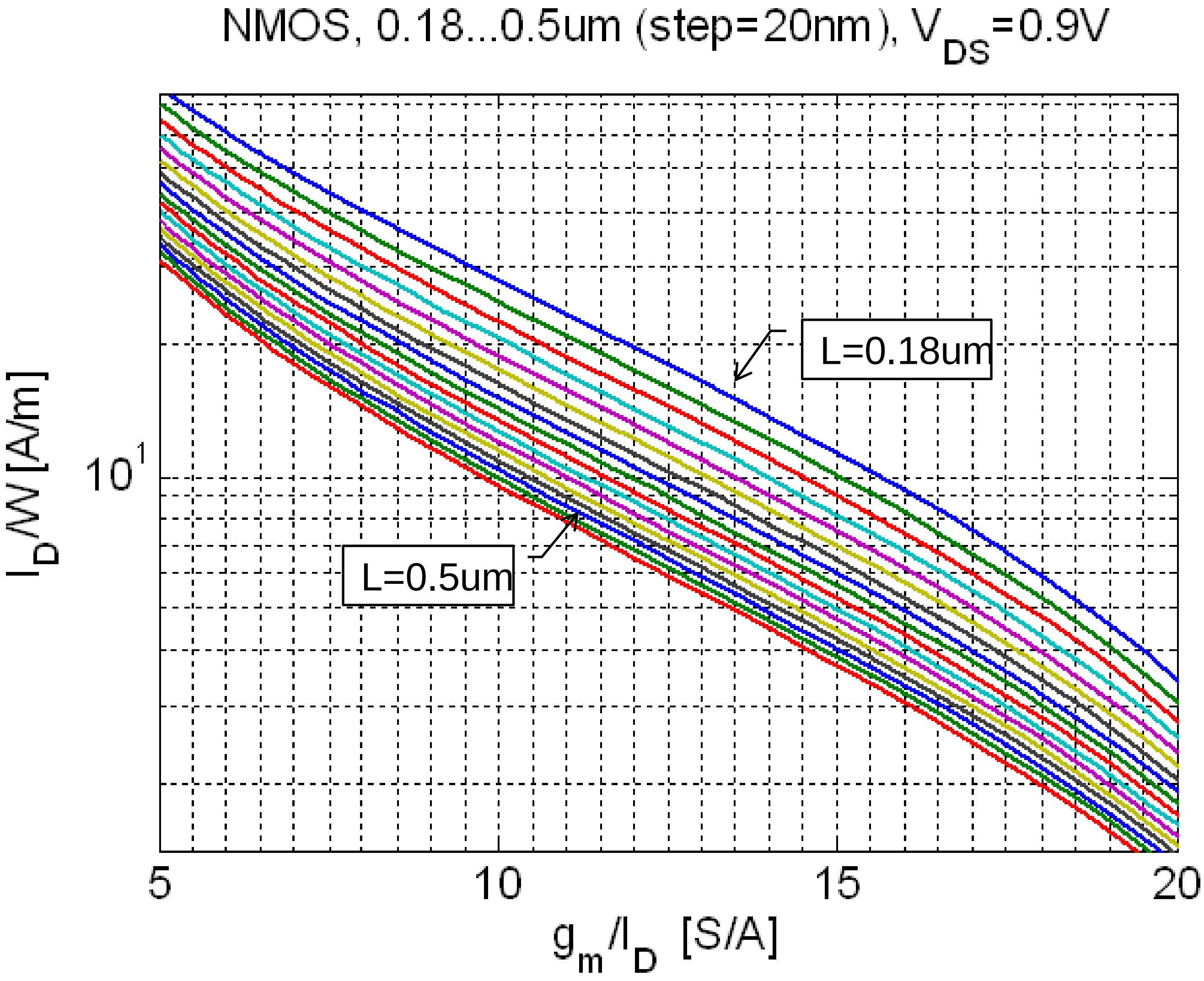
Transit Frequency Chart



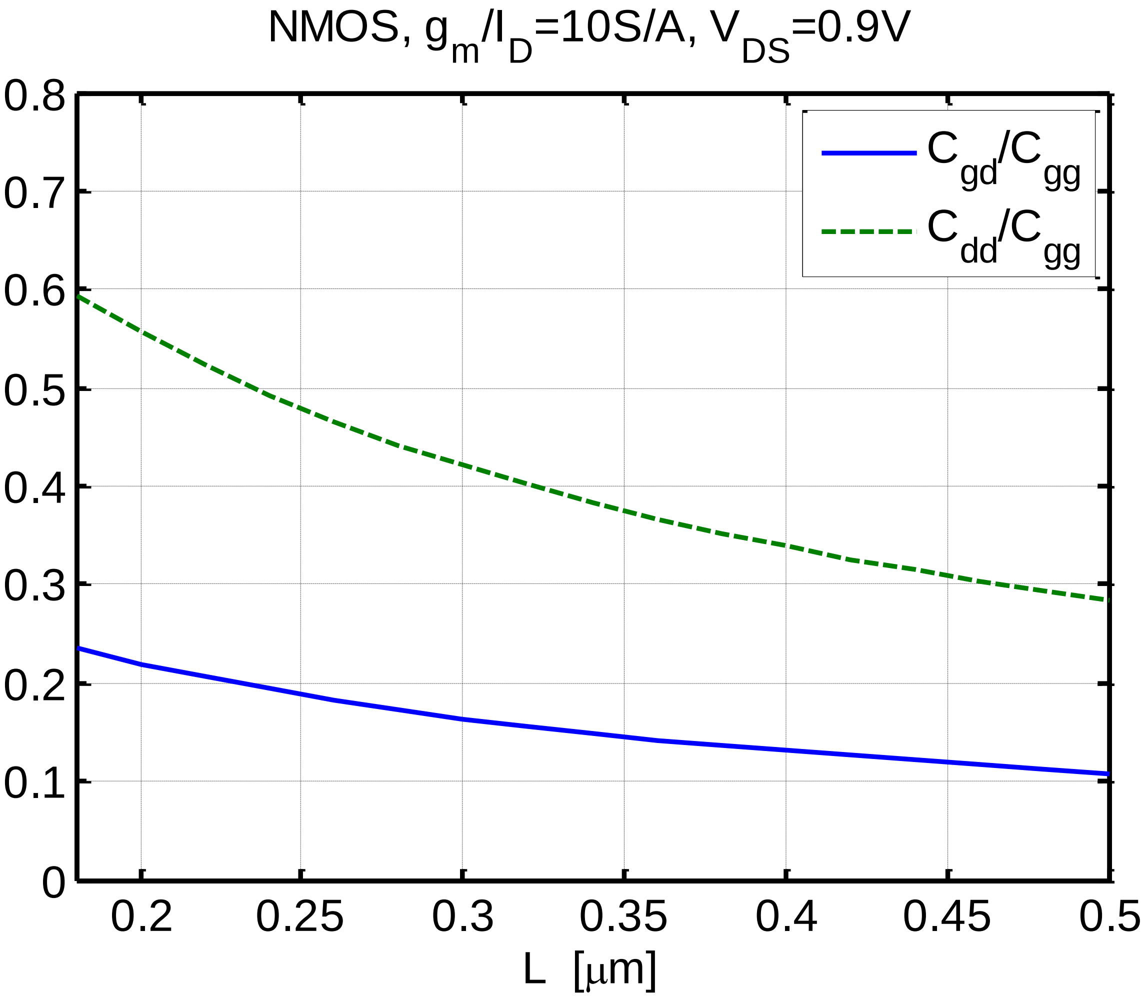
Intrinsic Gain Chart



Current Density Chart



Extrinsic Capacitances

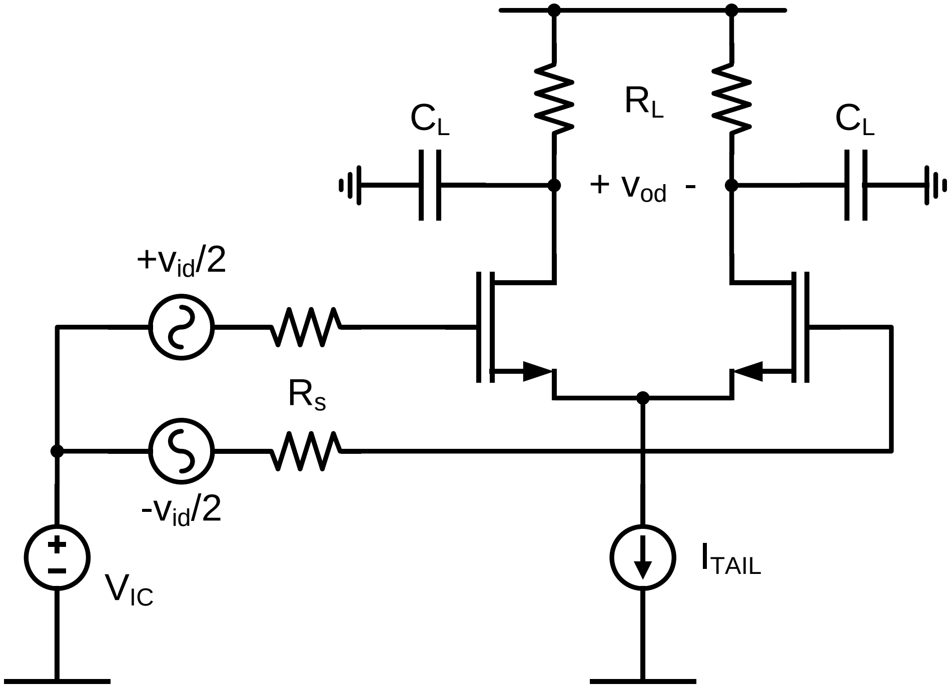


Generic Design Flow

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_{Dsat})
   - 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)

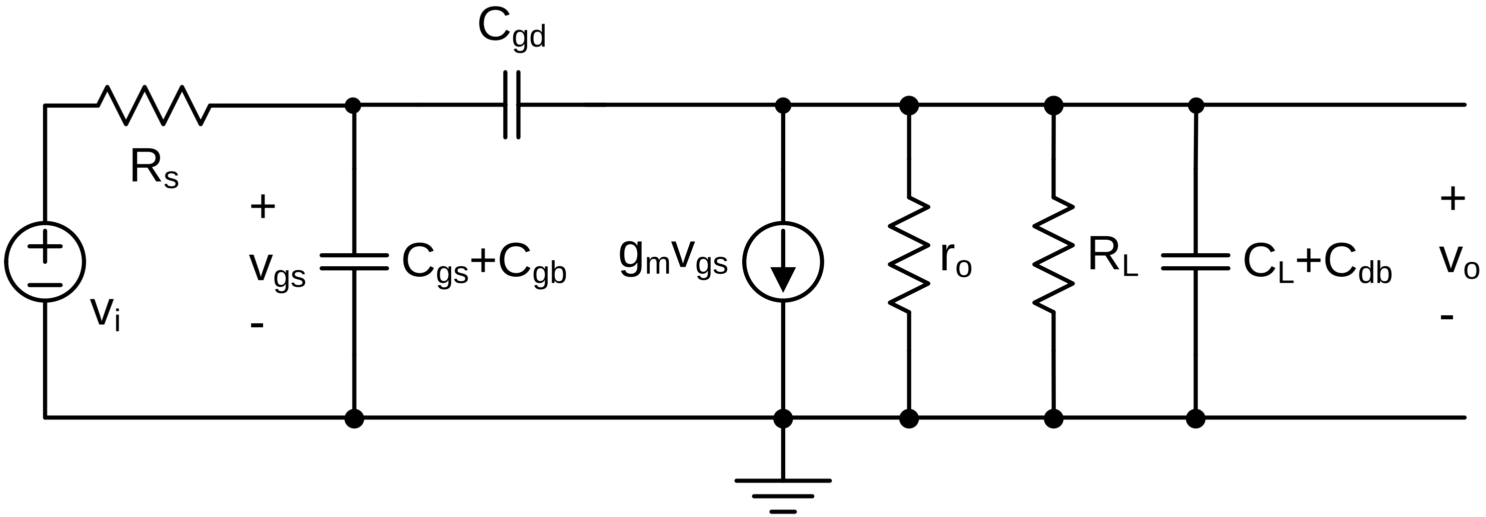
Many other possibilities exist (depending on circuit specifics, design constraints and objectives)

Back to Our 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 $I_{TAIL} \le 600\mu A$
    - 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

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

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

Zero and Pole Frequencies

High frequency zero (negligible)

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

Denominator coefficients

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

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

(C_{db} can be added to C_L if significant))

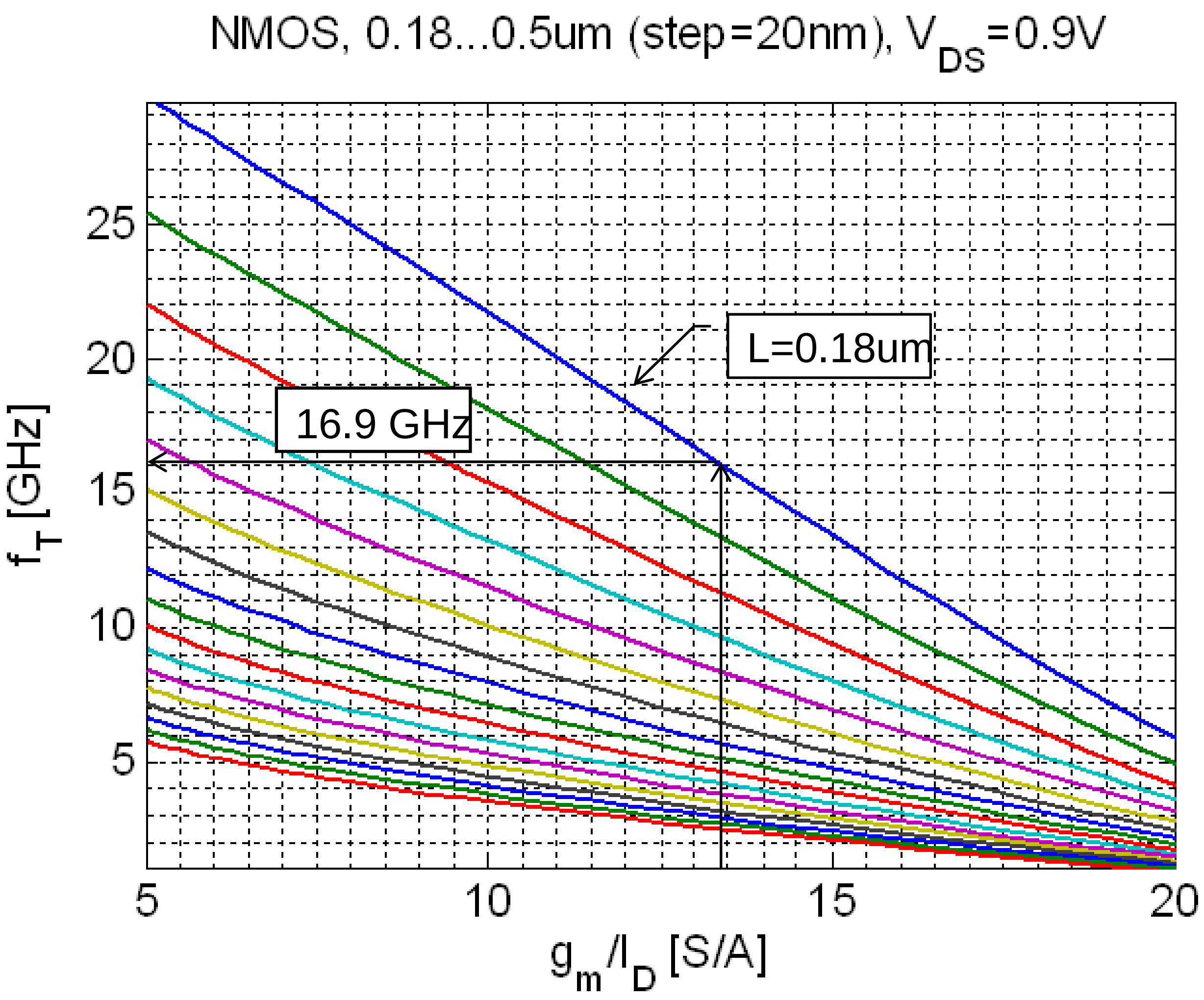
Dominant pole

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

Nondominant pole

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

Determine C_{gg} via f_T Look-up



L=0.18um

16.9 GHz

Find Capacitances and Plug in

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

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

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

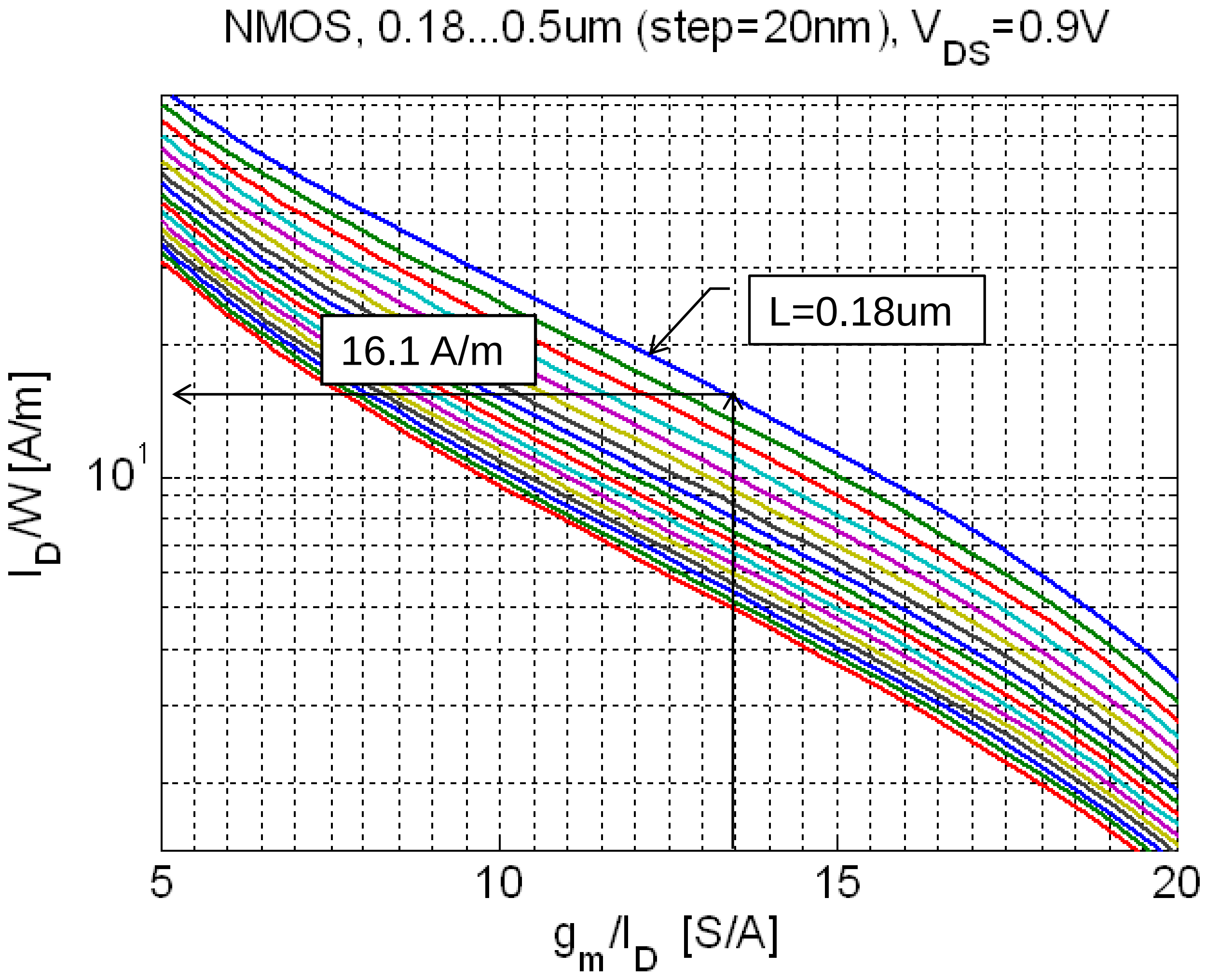
$$\mathbf{C_{db}=C_{dd}-C_{gd}=13.6fF}$$

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

$$\boxed{\mathbf{f_{p1}\cong 200\ MHz}}$$

$$\boxed{\mathbf{f_{p2}\cong 5.8\ GHz}}$$

Device Sizing



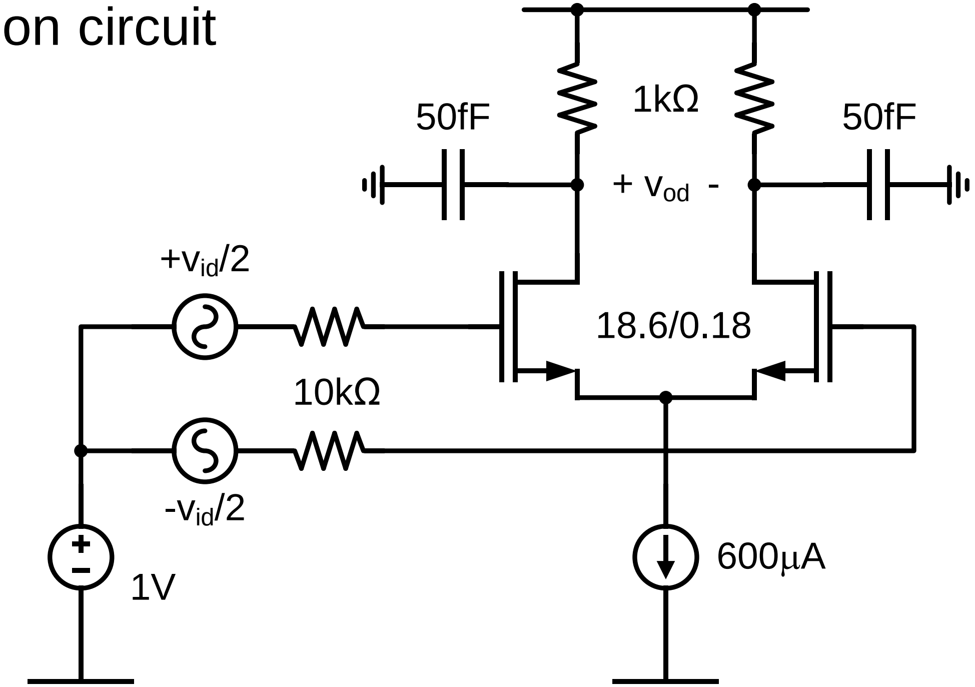
Matlab Design Script

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



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

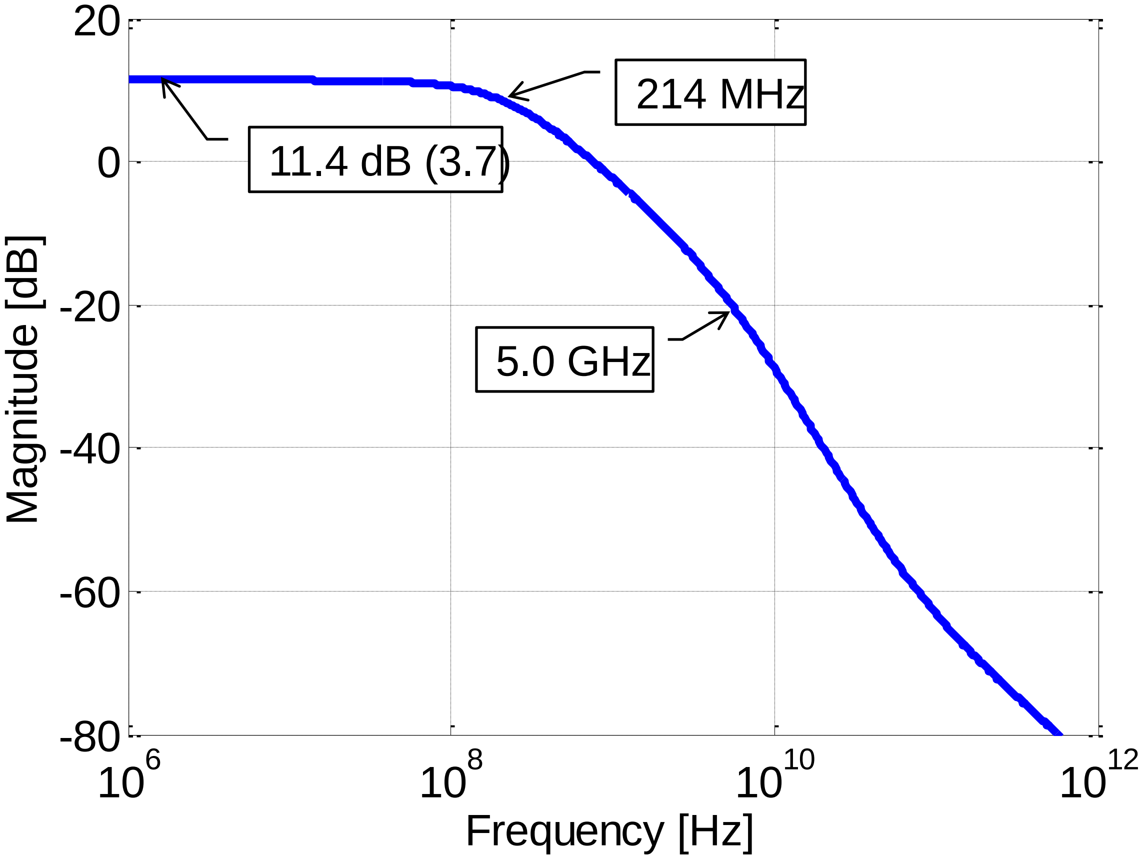
g_m = 4 mS

C_{dd} = 22.6 fF

C_{gg} = 37.8 fF

C_{gd} = 9.0 fF

Simulated AC Response

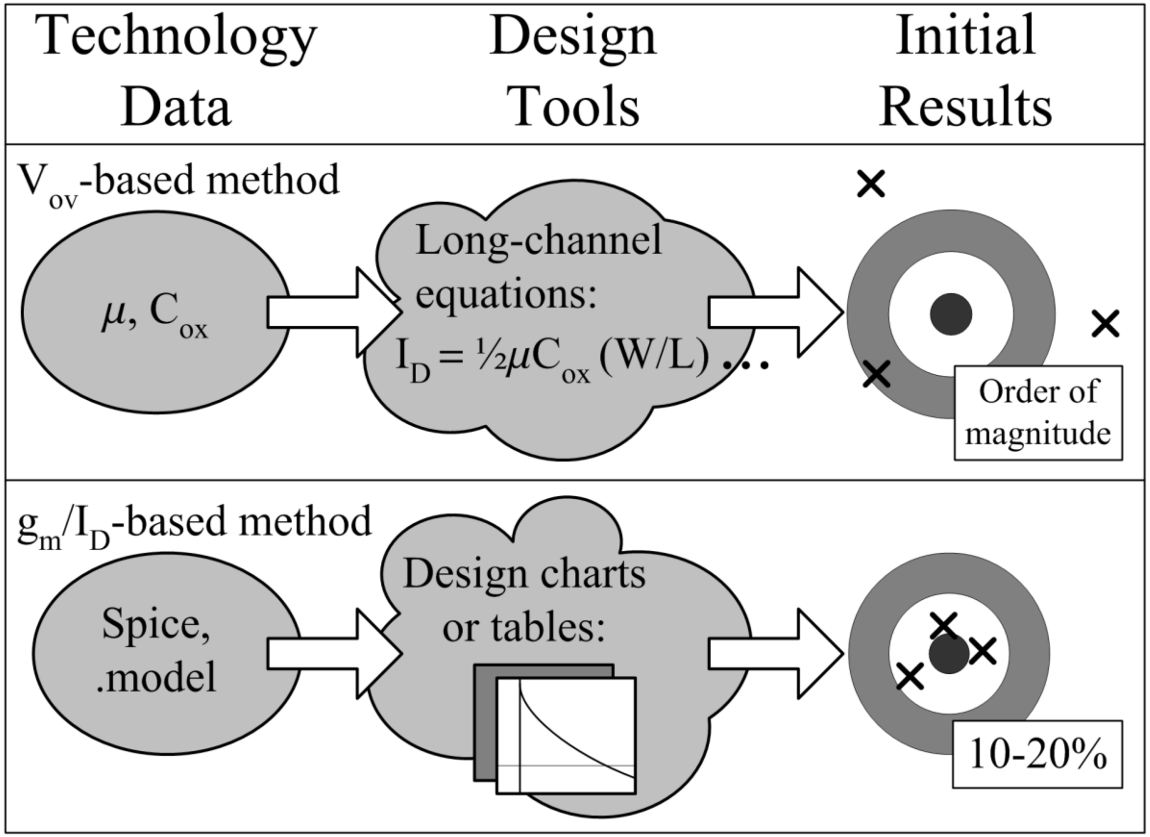


- Calculated values: $|A_{v0}|$=12 dB (4.0), f_{p1} = 200 MHz, f_{p2}= 5.8 GHz

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


Technology Data
Design Tools
Initial Results
V_{ov}-based method
μ, C_{ox}
Long-channel equations:
$I_D = ½\mu C_{ox} (W/L) \ldots$
Order of magnitude
g_m/I_D-based method
Spice, .model
Design charts or tables:
10-20%

References

- F. Silveira et al. "A gm/ID 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 gms/Id," 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 Circuits, 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 gm/ID Lookup Table Methodology,” IEICE Trans. Electronics, Vol. E94-C, No.3, Mar. 2011.

Summary on G_m/I_D-Based Design

- Think g_m/I_D!
- Weak inversion – moderate inversion – Strong inversion

  >20S/A … 15 S/A … <10 S/A

- g_m/I_D shows up naturally in many circuit calculations and provides a nice link between the most important small signal and large signal parameters
- Once g_m/I_D has been picked, look up current density and size transistor → Easy
- The hard part is to figure out the best inversion levels to use for the given specs → Clever analysis/scripting is needed…
- When in doubt, start with moderate inversion, often the best of both worlds (compromise between speed and efficiency)