Chapter 5
g_m/I_D - Based Design

Ross Walker
ECE/CS 5720/6720 Fall 2017
University of Utah
Partly adapted from Stanford's analog circuit design sequence

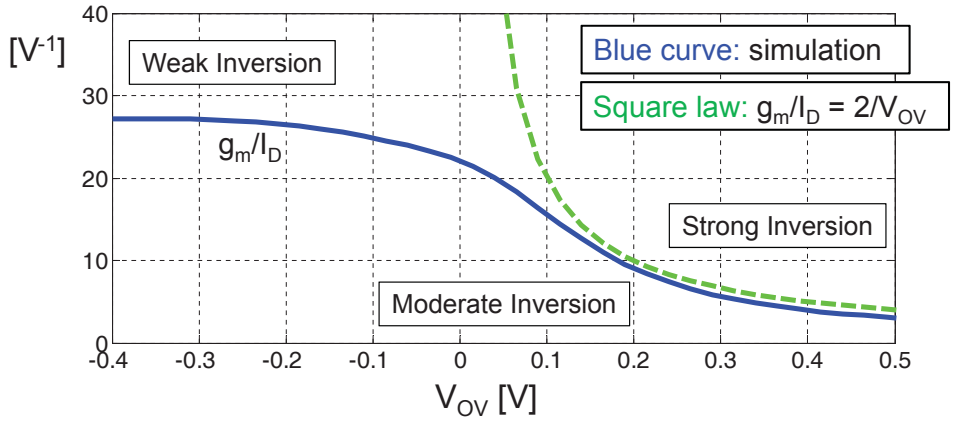
Reading:
See 'References' at the end of this chapter for optional reading.

Outline

- Review the three main figures of merit for transistors: g_m/I_D, ω_T, g_m/g_{ds}
  - Transistor characteristics that are directly linked to circuit performance
- See how the square law fails in practice
- Discuss g_m/I_D-based design using lookup tables/charts
  - Quantitative design methodology based on characterizing transistors
  - Intuitive framework for exploring and optimizing circuit performance
    - Transistor figures of merit become useful design variables
- Design example using g_m/I_D methodology
- Learning objectives
  - Understand tradeoffs between speed, power efficiency, and gain
    - In transistors, and in circuits
  - Be able to use technology characterization tables/charts to size transistors
    - Get practice in the design project
  - Develop a systematic design methodology in the absence of simple analytical models like the square law equation

FOM #1: g_m/I_D as a Figure of Merit

- "Transconductance efficiency" figure of merit for transistors
  - Circuit performance often dictates the required g_m
    - e.g. gain of a common source stage: $|A_v| = g_m R_L$
  - g_m/I_D quantifies how much g_m you get for the amount of bias current you invest
    - i.e. $g_m/I_D = 10S/A \rightarrow 10\mu S$ per $1\mu A$



Why Doesn't the Square Law "Work"?

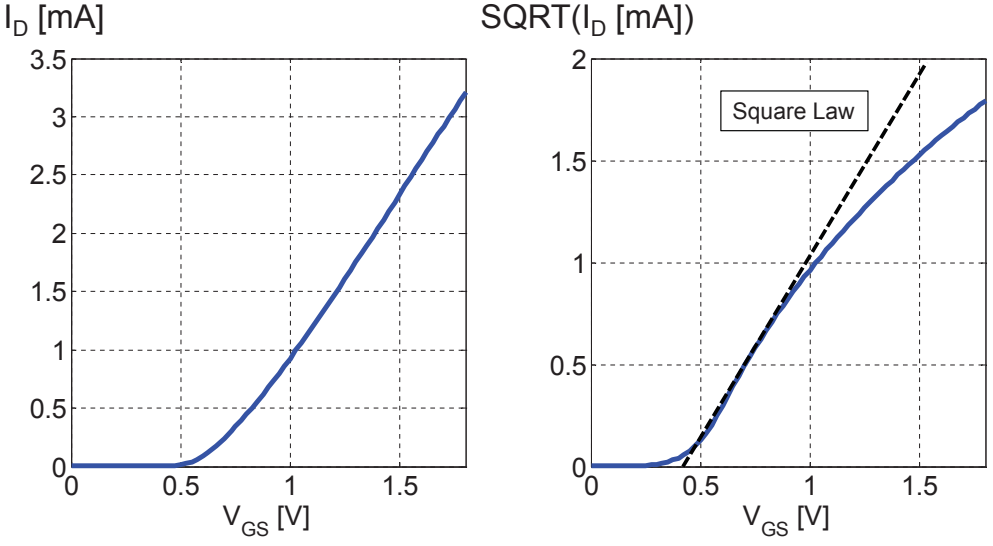
- MOSFETs are complicated!
- The IV-behavior in saturation can be roughly categorized according to the channel's inversion level: weak, moderate and strong inversion
- The current is due to diffusion in weak inversion and mostly due to drift in strong inversion; the transition is smooth and complicated
- 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

What are μC_{ox} ("KP") and λ ("LAMBDA") for our Technology?

```
.MODEL nmos6720 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
```

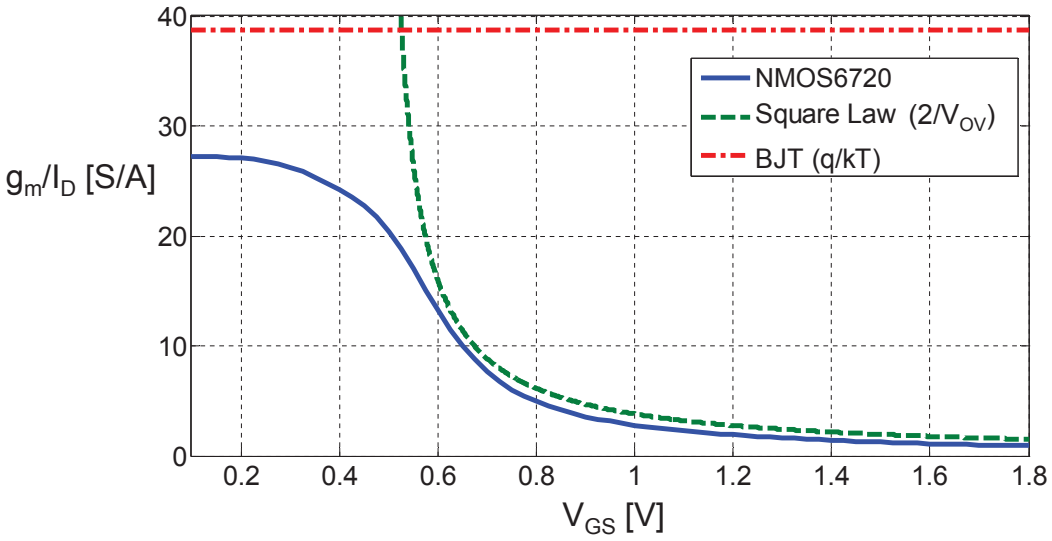
- The Spice model for an NMOS device in our course technology is shown to the left
- This is a 110-parameter BSIM3v3 model, quite complex even though 180nm is a relatively old process
  - More recent models may require even more parameters (e.g. PSP, BSIM6)
  - **KP and LAMBDA are nowhere to be found!**
- 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)



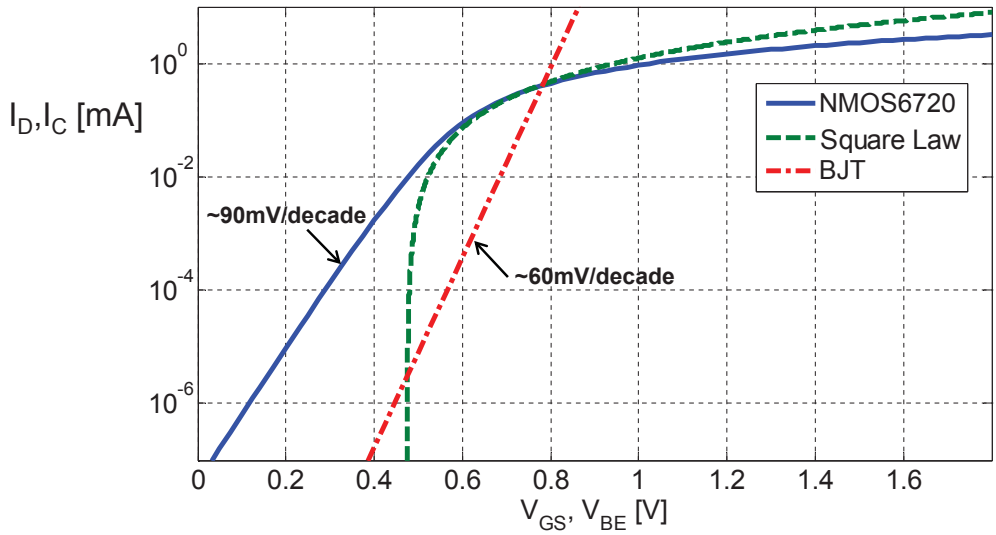
- Two observations
  - The transistor does not abruptly turn off at some V_t
  - The current is not perfectly quadratic with $(V_{GS}–V_t)$

g_m/I_D at Low V_{GS}



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

Currents on a Log Scale

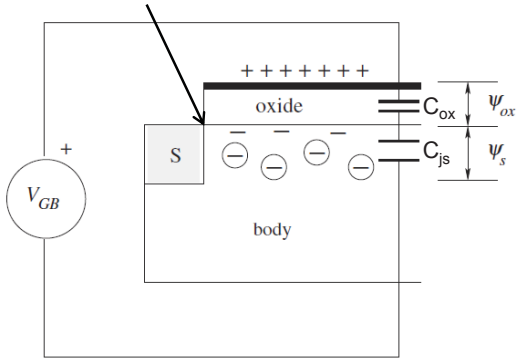


- What is V_t, anyway? The device does not turn off at all, but really approaches an exponential IV law for low V_{GS}
- What determines the current at low V_{GS}?

Weak Inversion (Subthreshold) Operation

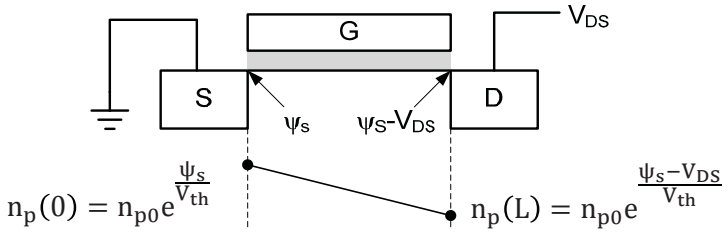
- Before inversion occurs, the electrostatic field from the gate forward-biases the source-side pn junction at the surface
- Physics governed by a "gated diode" model

Potential at this point is higher than body/source potential → forward bias



D.L. Pulfrey, Understanding Modern Transistors and Diodes, Cambridge University Press, 2010.

Resulting Diffusion Current



$$I_D = qAD_n \frac{n_p(0) - n_p(L)}{L}$$

$$I_D = \frac{1}{L} qAD_n n_{p0} e^{\frac{\psi_s}{V_{th}}} \left(1 - e^{-\frac{V_{DS}}{V_{th}}}\right)$$

Here v_{th} = kT/q is the "thermal voltage"

- The current grows exponentially with ψ_s
- The current becomes independent of V_{DS} for $V_{DS} > 3V_{th}$ (~78mV)

Capacitive Divider

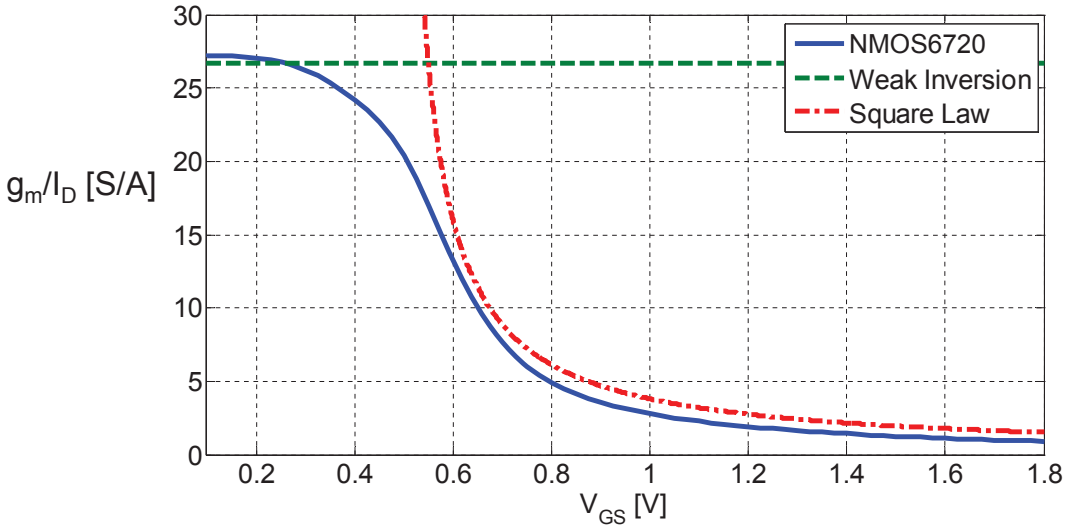
$$\frac{d\psi_s}{dV_{GS}} = \frac{C_{ox}}{C_{js} + C_{ox}} = \frac{1}{n}$$

- n is called “subthreshold factor” or “nonideality factor”
- n ≅ 1.45 for an NMOS device in the ECE6720 technology
- After including this relationship between ψ_s and V_{GS} and after a few additional manipulations, the final expression for the drain current becomes:

$$I_D = \frac{W}{L} I_{D0} e^{\frac{V_{GS}-V_t}{nV_{th}}} \left(1 - e^{-\frac{V_{DS}}{V_{th}}}\right)$$

where I_{D0} depends on technology (I_{D0} ≅ 0.43μA for an NMOS device in ECE6720 technology)

g_m/I_D



- We now have a better idea about the maximum possible g_m/I_D, but we still do not know how to model the transition region between the two IV laws

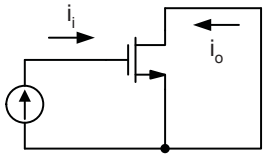
Moderate Inversion

- In the transition region between weak and strong inversion, the drain current consists of both drift and diffusion currents
- One can show that the ratio of drift/diffusion current in moderate inversion and beyond is approximately $(V_{GS}-V_t)/(kT/q)$
- This means that the square law equation (which assumes 100% drift current) does not work unless the gate overdrive is several kT/q

- Is there a simple expression that works for all three regions (weak, moderate and strong inversion)?
  - No, there is no closed-form expression that captures all modes of operation as well as “short channel effects”
  - This is why g_m/I_D-based design is very useful…

FOM #2: Transit Frequency

- Transistor figure of merit that relates to circuit speed and efficiency
  - The transit frequency of a transistor has "historically" been defined as the frequency where the magnitude of the common source current gain ($|i_o/i_i|$) falls to unity



(Biasing not shown)

- Ignoring extrinsic capacitance and C_{gb}, and using the square law model:

$$\omega_T = \frac{g_m}{C_{gs}} \cong \frac{3}{2}\frac{\mu V_{OV}}{L^2}$$

- We’ll use this improved definition for g_m/I_D-based design:

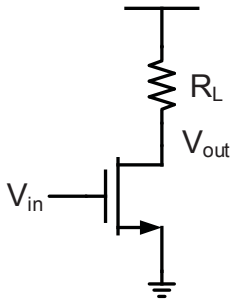
$$\omega_T \triangleq \frac{g_m}{C_{gg}} = \frac{g_m}{C_{gs}+C_{gb}+C_{gd}}$$

Transit Frequency Interpretation

- The transit frequency is only useful as a figure of merit in that it quantifies g_m/C_{gg}
  - Tells you how much C_{gg} you get for a given g_m
- It does **not** accurately predict up to which frequency you can use the device
  - See slide 46 of Chapter 2
- At a high level, C_{gg} affects circuit speed
  - At a lower level, C_{gs}, C_{gb}, and C_{gd} affect a circuit's bandwidth differently
    - e.g. Miller effect in a common course stage, $(1+|Av|)C_{gd}$ versus C_{gs}
- We'll see that C_{gs}/C_{gg}, C_{gd}/C_{gg}, etc. are fairly stable ratios
  - Having a small C_{gg} means the individual components are small too

FOM #3: Intrinsic Gain

- Transistor figure of merit related to the maximum gain a circuit can achieve



$$\frac{V_{out}}{V_{in}} = -g_m(r_o||R_L)$$

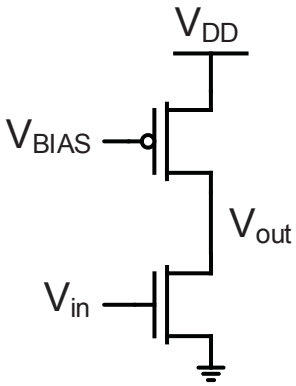
$$\max\left(\left|\frac{V_{out}}{V_{in}}\right|\right) = g_m r_o = \frac{g_m}{g_{ds}}$$

Square law: $g_m r_o = \frac{1}{\lambda}\frac{g_m}{I_D} = \frac{1}{\lambda}\frac{2}{V_{OV}}$ } A *very* poor model

- Intrinsic gain is typically related to circuit accuracy
  - e.g. in feedback circuits, we use high open-loop gain to make an accurate closed-loop gain

Example: Intrinsic Gain and Active Loads

- The gain of amplifiers with active loads is fundamentally linked to intrinsic gain



$$\left|\frac{V_{out}}{V_{in}}\right| = g_{mn}\left(r_{on}||r_{op}\right) = g_{mn}r_{on} \;||\; g_{mn}r_{op}$$

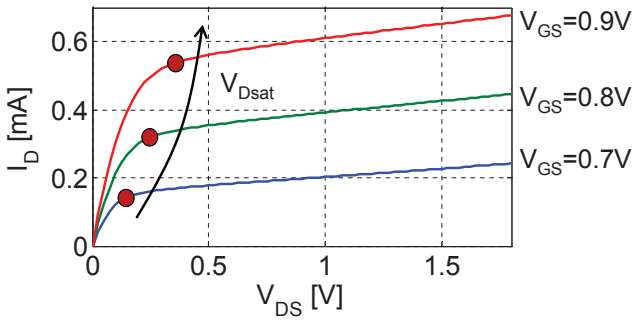
$$\ldots \quad = (g_m r_o)_n \;||\; \frac{(g_m/I_D)_n}{(g_m/I_D)_p}(g_m r_o)_p$$

- The ratio of g_{mp}/g_{mn} is usually designed to be low, for low noise
  - Generally constrained by practical biasing ($V_{Dsat} \approx V_{OV}$)

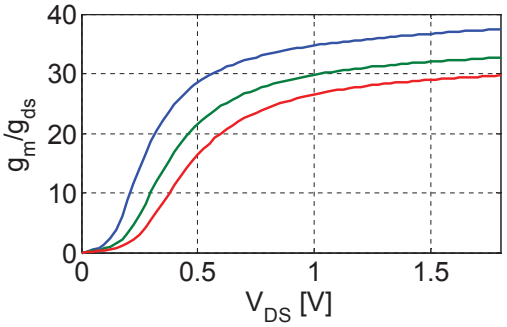
$\frac{(g_m/I_D)_n}{(g_m/I_D)_p}$ is typically 2-4

How about V_{Dsat}?

- V_{Dsat} tells us how much voltage we need across the channel to operate in saturation
  - "High gain region"



- It is important to note that V_{Dsat} is not crisply defined in modern devices
  - g_m/g_{ds} gradually increases 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

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

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

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

Consistent with the classical first-order relationship

Weak Inversion

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

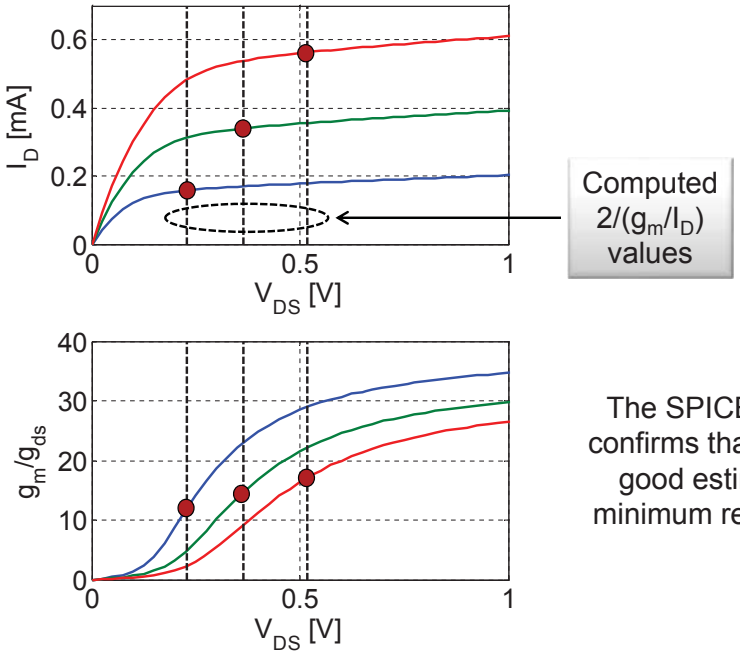
Need about $3V_{th}$ for saturation

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

$$\frac{2}{(g_m / I_D)} = 2nV_{th} \cong 3V_{th}$$

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}

Summary: Transistor Figures of Merit for Design

Square Law

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

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

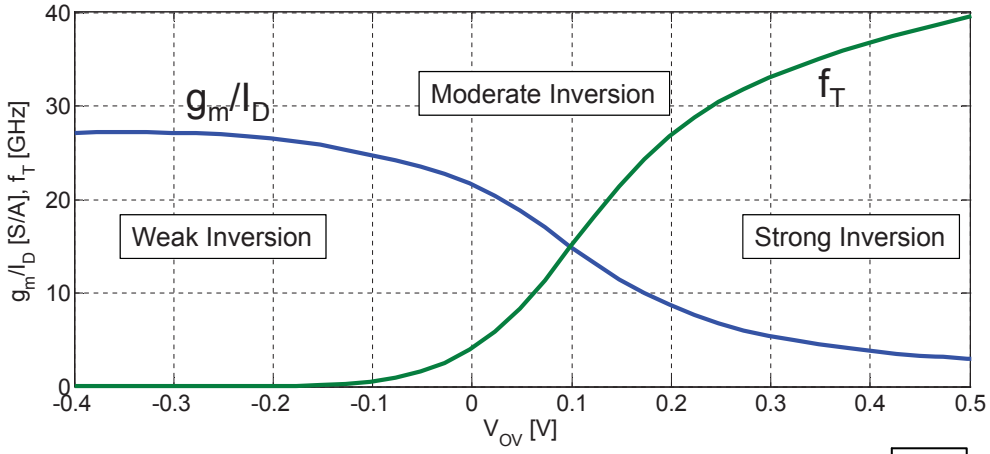
- Transit frequency
  - Want large g_m, without large C_{gg}

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

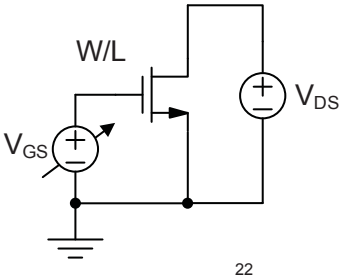
- Intrinsic gain
  - Want large g_m, but no g_{ds}

$$\frac{g_m}{g_{ds}} \cong \frac{2}{\lambda V_{OV}}$$

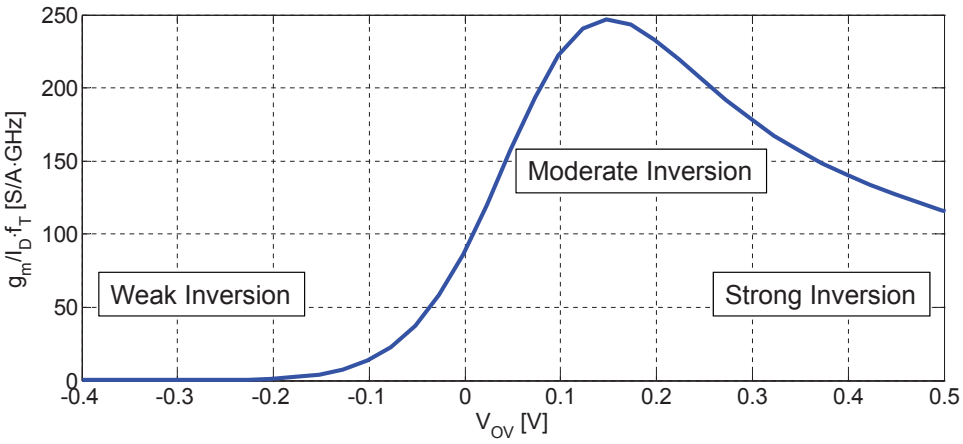
Design Tradeoff: g_m/I_D and f_T



- 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

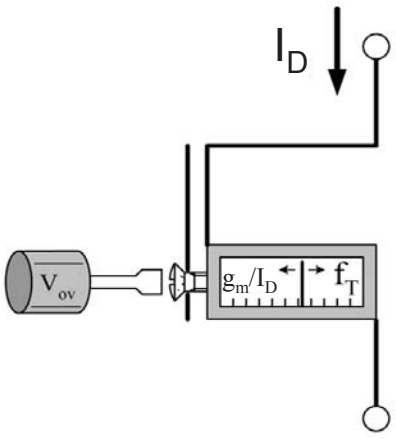


Product of g_m/I_D and f_T



- Interestingly, the product of g_m/I_D and f_T peaks in moderate inversion
  - Peaks around $g_m/I_D \approx 10$S/A in ECE6720 technology
- Operating the transistor in moderate inversion makes sense when we value speed and power efficiency equally
  - This is just a heuristic, not always the case

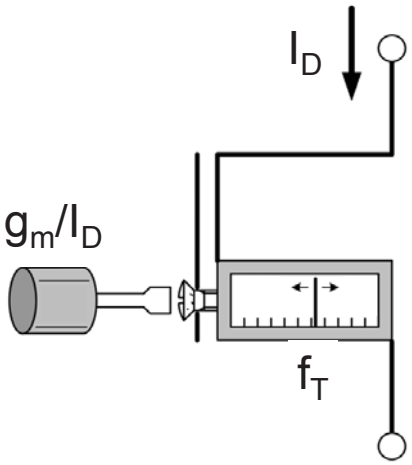
Design in a Nutshell

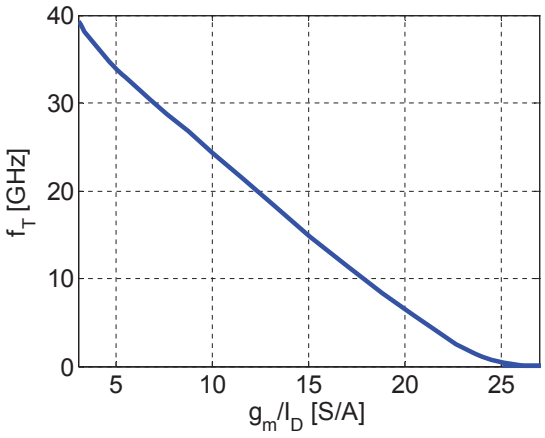


- Choose length such that the circuit has 'enough' gain
- Choose the inversion level according to the proper tradeoff between speed (f_T) and transconductance efficiency (g_m/I_D) for the given circuit
- The inversion level is fully determined by the gate overdrive V_{OV}
  - But, V_{OV} is not a very 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}

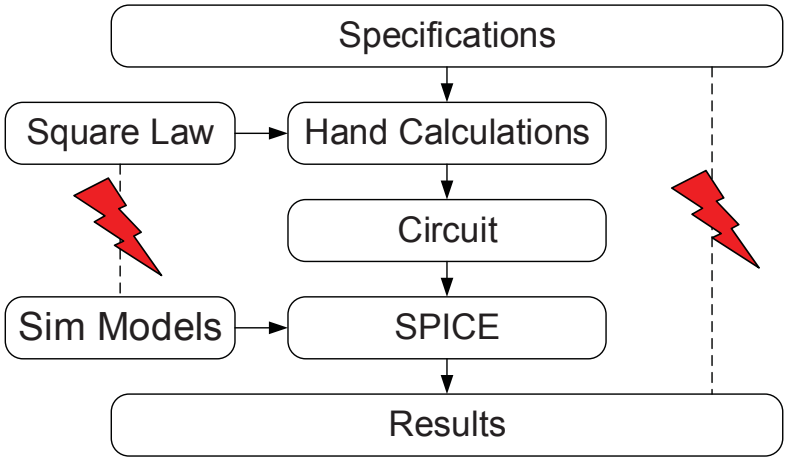




Outline

- Review the three main figures of merit for transistors: g_m/I_D, ω_T, g_m/g_{ds}
  - Transistor characteristics that are directly linked to circuit performance
- See how the square law fails in practice
- Discuss g_m/I_D-based design using lookup tables/charts
  - Quantitative design methodology based on characterizing transistors
  - Intuitive framework for exploring and optimizing circuit performance
    - Transistor figures of merit become useful design variables
- Design example using g_m/I_D methodology
- Learning objectives
  - Understand tradeoffs between speed, power efficiency, and gain
    - In transistors, and in circuits
  - Be able to use technology characterization tables/charts to size transistors
    - Get practice in the design project
  - Develop a systematic design methodology in the absence of simple analytical models like the square law equation

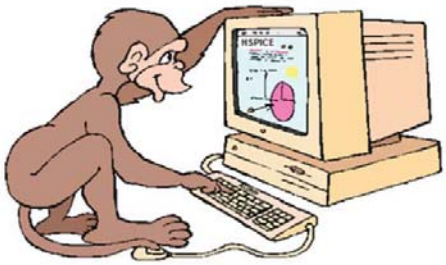
The Problem



- 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

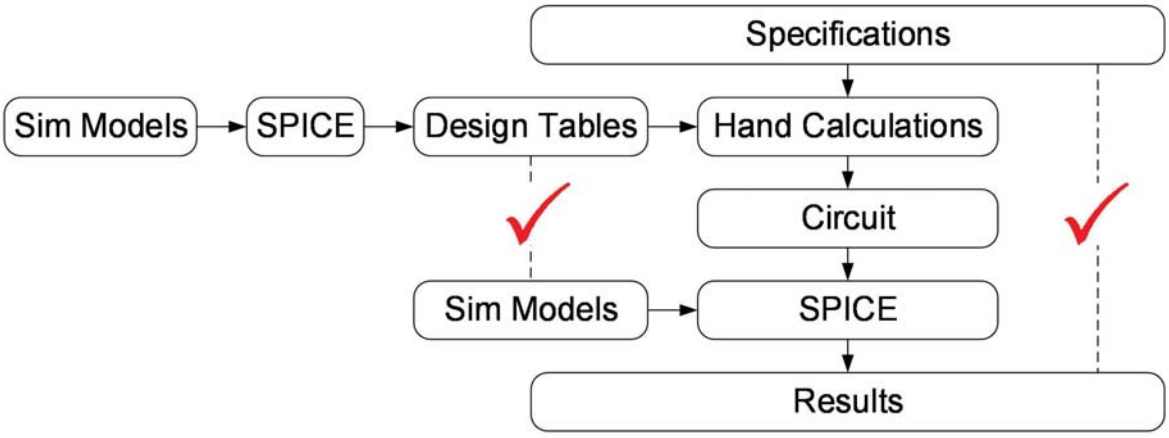
Unfortunate Consequence

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



[Courtesy Isaac Martinez]

The Solution



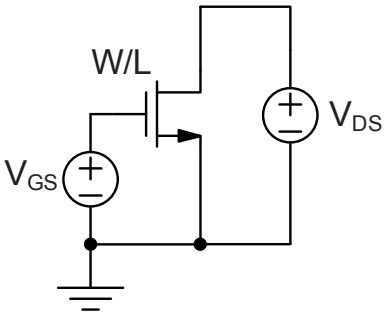
- Use pre-computed SPICE data in hand calculations

g_m/I_D-centric Technology Characterization

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

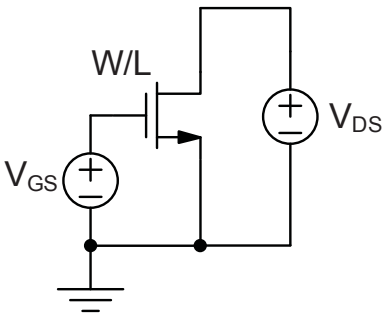
Starting Point: Technology Characterization via DC Sweep

- Obtain these tables through a DC sweep simulation of the transistor models
  - Measure transistor .op parameters at each point of the sweep
    - g_m, I_D, C_{gg}, g_{ds}, etc.
  - Repeat the sweep for different lengths
    - 180nm, 200nm, ….. 3µm
  - Generate charts, Matlab arrays, etc.
- Simple version: sweep V_{GS} with V_{DS} held fixed at $V_{DS} = V_{DD}/2$
  - The figures of merit and I_D/W don't vary too much with V_{DS}
- Advanced version: sweep V_{DS} and V_{BS} also



HSPICE Example

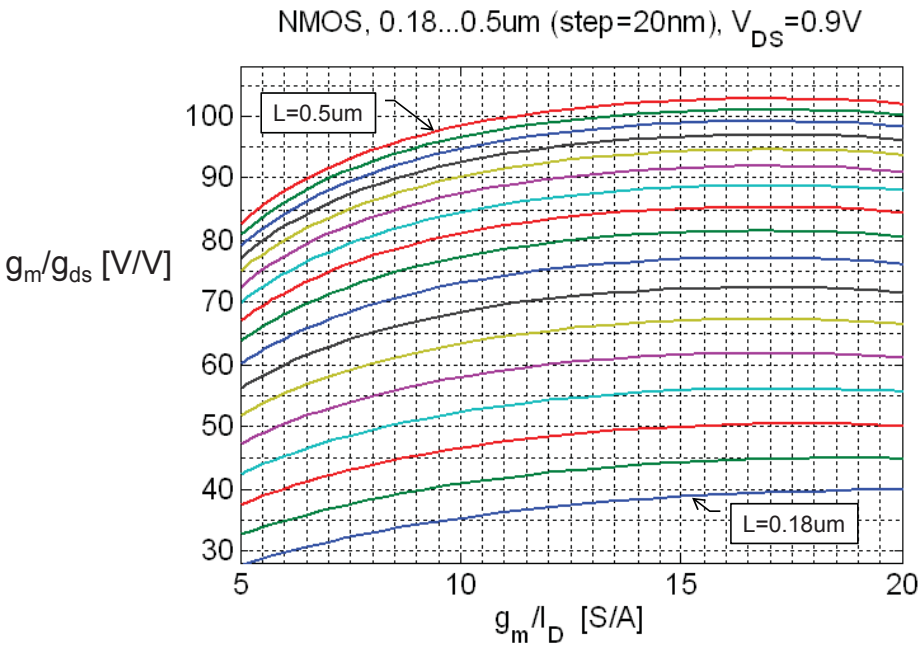
```
* Example HSPICE netlist/simulation, techchar.sp
* Model file
.inc './models/ece6720.mod'
* Define 'width' and 'length' in a parameter file,
* along with 'gsmax' and 'gsstep'
.inc 'techchar_params.sp'

.param ds = 0.9
.param gs = 0.9

vdsn     vdn 0         dc 'ds'
vgsn     vgn 0         dc 'gs'
mn       vdn vgn 0 0 nmos6720 L='length' W='width'

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

.probe n_id   = par('i(mn)')
.probe n_gm   = par('gmo(mn)')
.probe n_gds  = par('gdso(mn)')
.probe n_cgg  = par('cggbo(mn)')
```



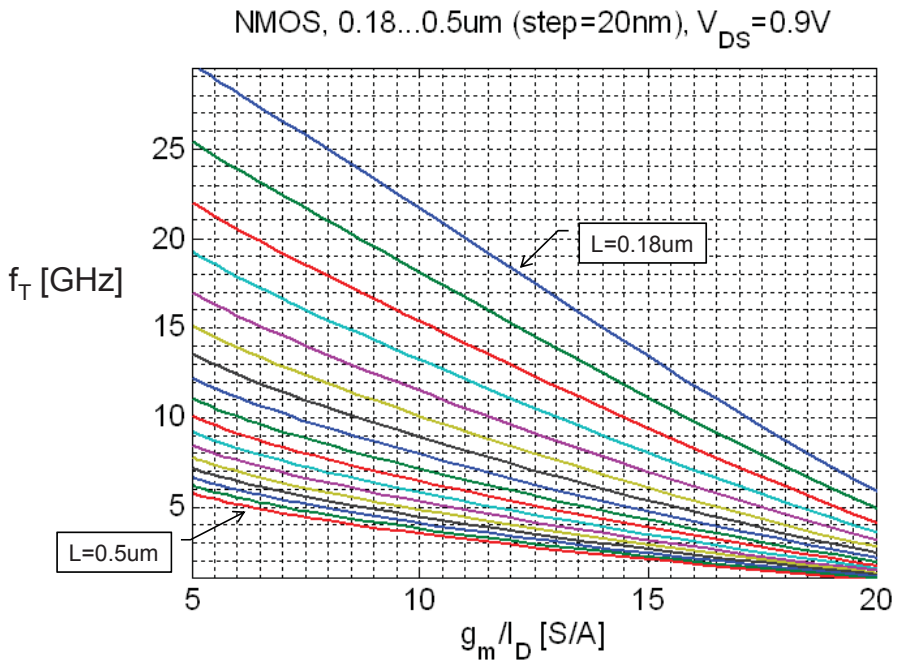
Example Matlab Wrapper

```
% Example Matlab wrapper

% HSPICE toolbox
addpath('./HSPICEToolBox')

% Parameters for HSPICE runs
VGS_step = 25e-3; VGS_max = 1.8;
VGS = 0:VGS_step:VGS_max;
W = 5e-6; L = [(0.18:0.02:0.5) (0.6:0.1:1.0) (1.2:0.2:3)]*1e-6;

% HSPICE simulation loop
for i = 1:length(L)
    % 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);
    fprintf(fid,'.param length = %d\n', L(i));
    fprintf(fid,'.param gsstep = %d\n', VGS_step);
    fprintf(fid,'.param gsmax  = %d\n', VGS_max);
    fclose(fid);
    system('/uusoc/facility/cad_common/Synopsys/hspice_G-2012.06-SP1/hspice/bin/hspice...
            techchar.sp >! techchar.out');
    %Read and store results
    h = loadsig(techchar.sw0);
    nch.GM(i,:) = evalsig(h, 'n_gm');
    nch.ID(i,:) = evalsig(h, 'n_id');
    nch.CGG(i,:) = evalsig(h, 'n_cgg');
    nch.GDS(i,:) = evalsig(h, 'n_gds');
end
```

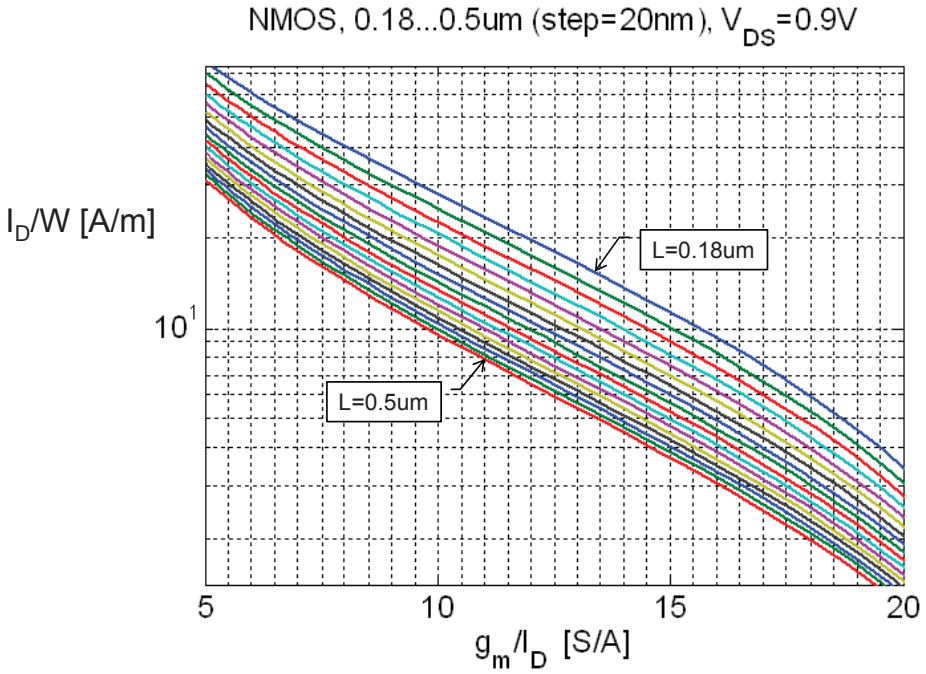

Intrinsic Gain Chart



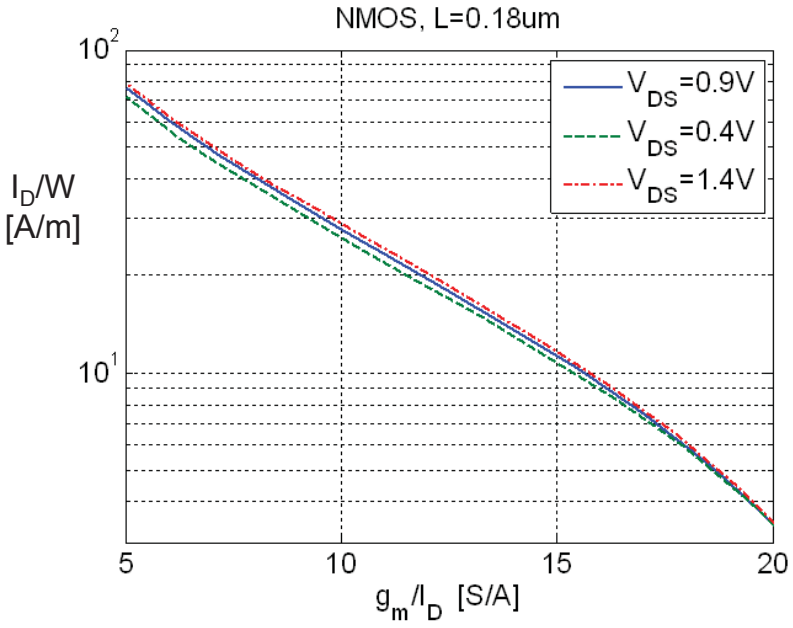
Transit Frequency Chart



Current Density Chart

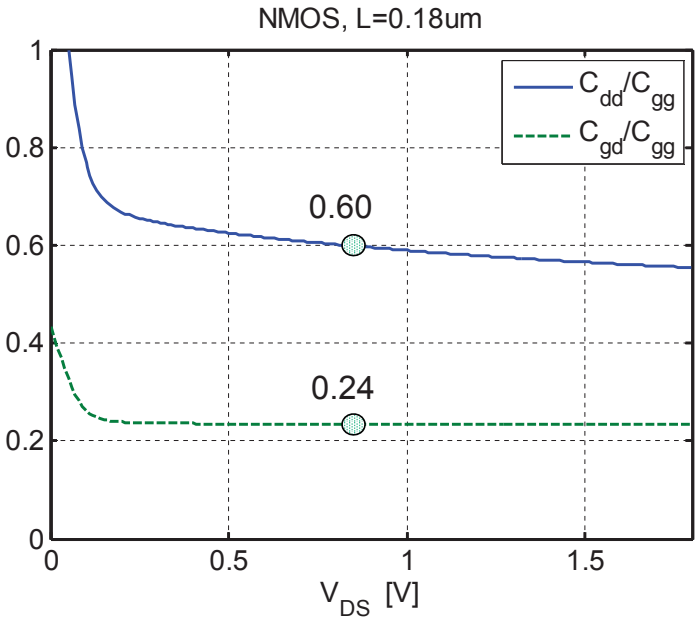


V_{DS} Dependence



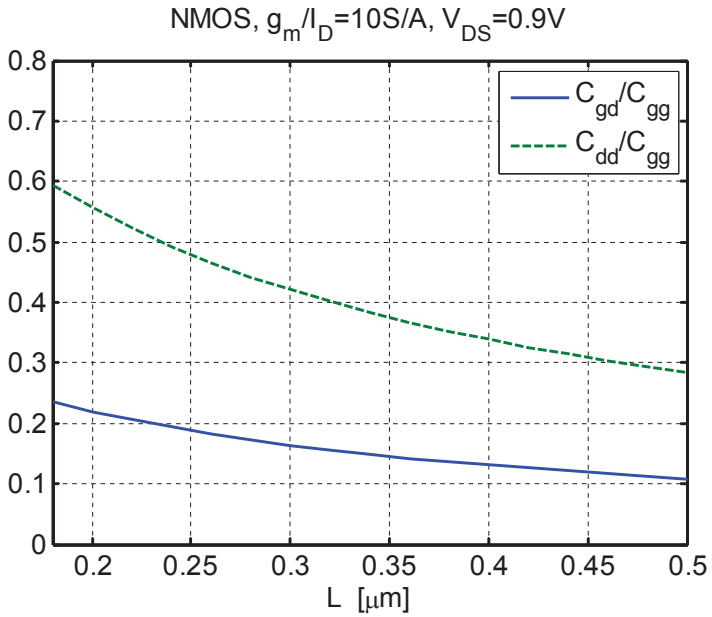
- V_{DS} dependence is relatively weak
- Typically OK to work with data generated for $V_{DD}/2$

Handling Extrinsic Capacitances

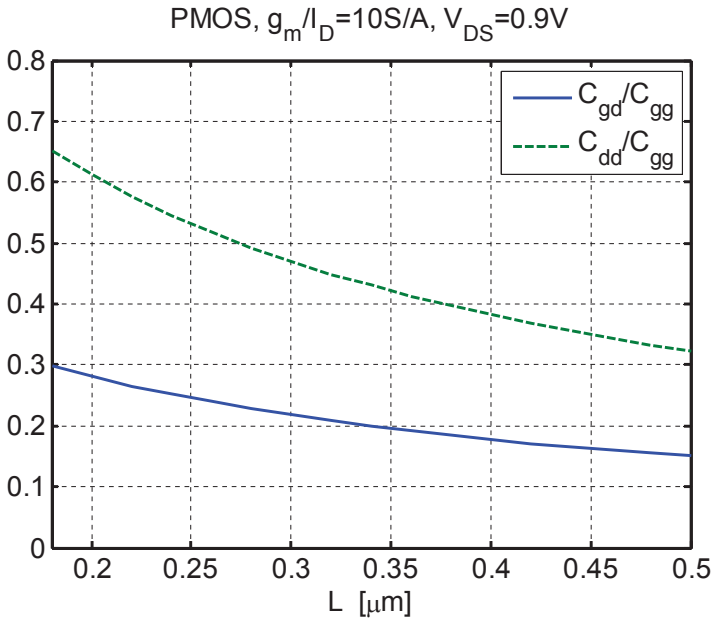


- Again, it's usually OK to work with estimates taken at $V_{DS}=V_{DD}/2$

Extrinsic Capacitances – Length Dependence



Extrinsic Capacitances (PMOS) – Length Dependence



Key Question

- How can we use all this data for **systematic** design?

- Many options exist
  - And you can invent your own, if you like

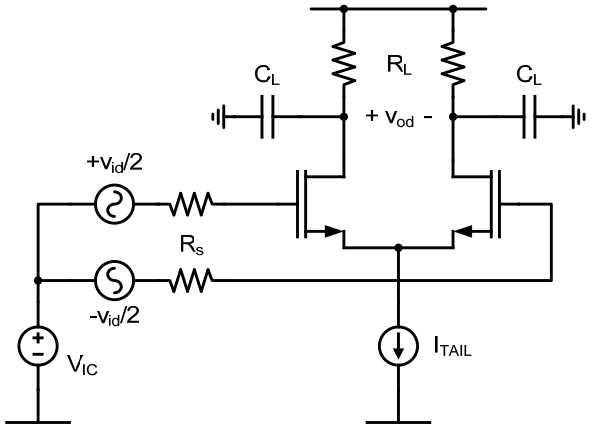
- Method taught in ECE6720
  - Look at the transistor in terms of width-independent figures of merit that are intimately linked to design specifications and performance
    - Rather than physical modeling parameters that do not directly relate to circuit specs
  - Think about the design tradeoffs in terms of the MOSFET's inversion level (bias point), using g_m/I_D as a proxy

A 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
   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)

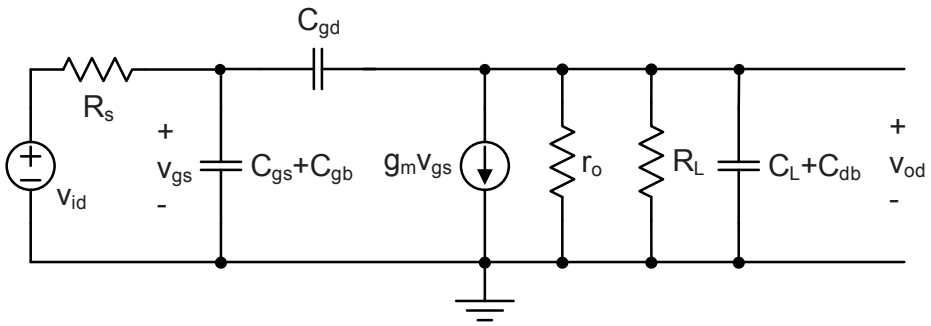
Many other possibilities exist (depends on circuit specifics, design constraints and objectives)

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

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

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

Aside: Why can we Neglect r_o?

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

$$\therefore\ 4 = g_m R_L \,||\, g_m r_o$$

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

- Even at $L=L_{min}= 0.18\mu m$, we have $g_m r_o > 30$

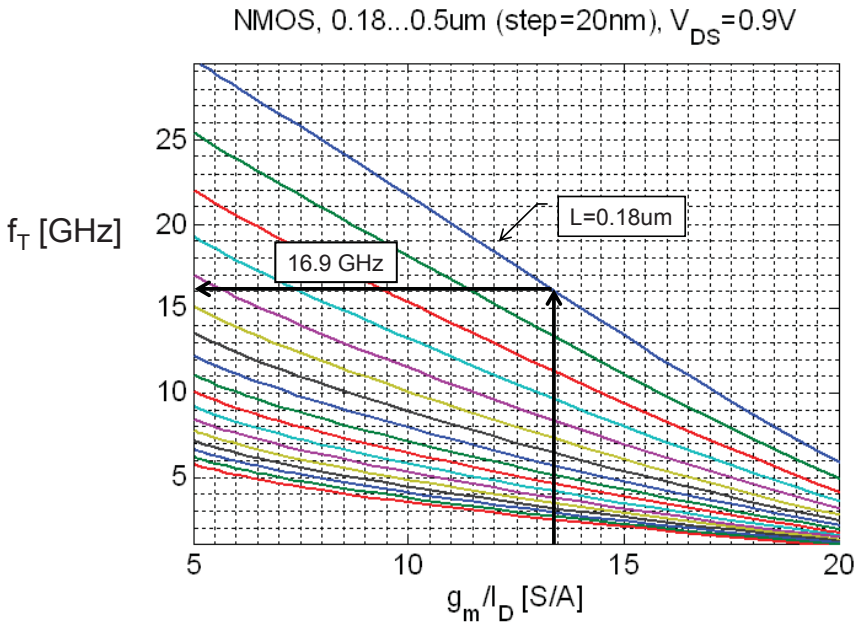
$$\frac{1}{4} >> \frac{1}{30}$$

$$\therefore\ \frac{1}{4} \cong \frac{1}{4} - \frac{1}{g_m r_o} = \frac{1}{g_m R_L}$$

$$4 \cong g_m R_L$$

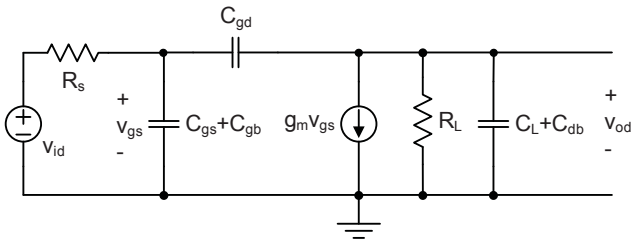
- r_o is negligible in this design problem

Determine C_{gg} via f_T Look-up



$$\frac{g_m}{I_D} = 13.3\frac{S}{A}$$

Calculate Capacitance Values



$$\mathbf{C_{gg} = \frac{1}{2\pi}\frac{4mS}{16.9GHz} = 37.7fF}$$

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

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

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

$$\mathbf{C_{gs} + C_{gb} = C_{gg} - C_{gd} = 28.7fF}$$

Zero and Pole Expressions

KCL analysis shows you: $H(s) = \dfrac{-g_m R_L (1 - sC_{gd}/g_m)}{1 + b_1 s + b_2 s^2}$

High frequency zero: $\omega_z = \dfrac{g_m}{C_{gd}} >> \omega_T \rightarrow$ negligible

Denominator coefficients:

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

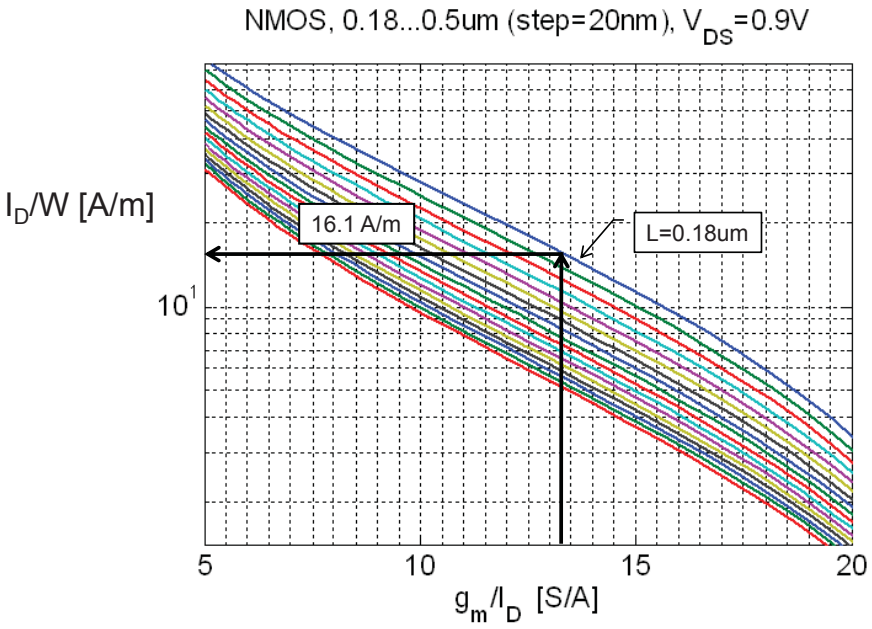
$$b_2 = R_s R_L (C_{gs}C_L + C_{gs}C_{gd} + C_L C_{gd})$$

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

Dominant pole approximation: $\omega_{p1} \cong \dfrac{1}{b_1}$ $\boxed{\mathbf{f_{p1} \cong 200\ MHz}}$

Non-dominant pole approximation: $\omega_{p2} \cong \dfrac{b_1}{b_2}$ $\boxed{\mathbf{f_{p2} \cong 5.8\ GHz}}$

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 ~ W/W
    - g_m/I_D ~ W/W
  - There is a one-to-one mapping from g_m/I_D to current density

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

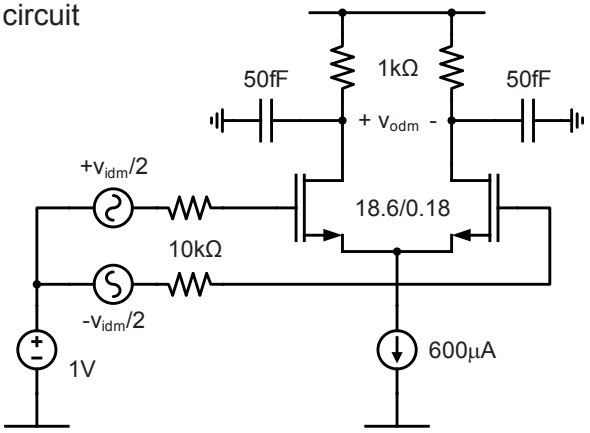
General case: $\frac{g_m}{I_D} = f(V_{OV})$ $\frac{I_D}{W} = g(V_{OV}) = g\left(f^{-1}\left(\frac{g_m}{I_D}\right)\right)$

Circuit For SPICE Verification

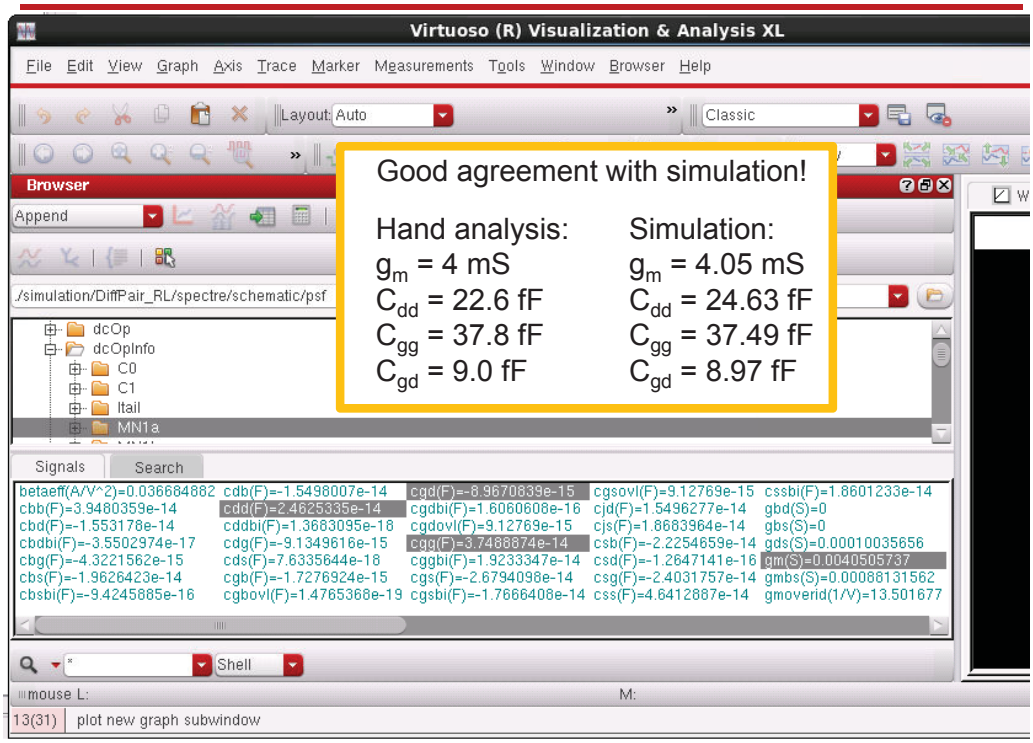
- Device width

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

- Simulation circuit



Simulated DC Operating Point (Cadence/Spectre)



Spectre dcOpInfo Capacitance Output Parameters

Spectre dcOpInfo output parameter

cdd	24.6253f
cgg	37.4887f
css	46.4128f
cbb	39.4803f
cgs	-26.7941f
cgd	-8.9671f

Note: you can ignore the negative signs, they are artifacts from the way Spectre computes small signal capacitances, e.g. $C_{gs}=dQ_g/dV_s$

Corresponding Small Signal Model Elements

cdd $\equiv C_{gd} + C_{db}$

cgg $\equiv C_{gs} + C_{gd} + C_{gb}$

css $\equiv C_{gs} + C_{sb}$

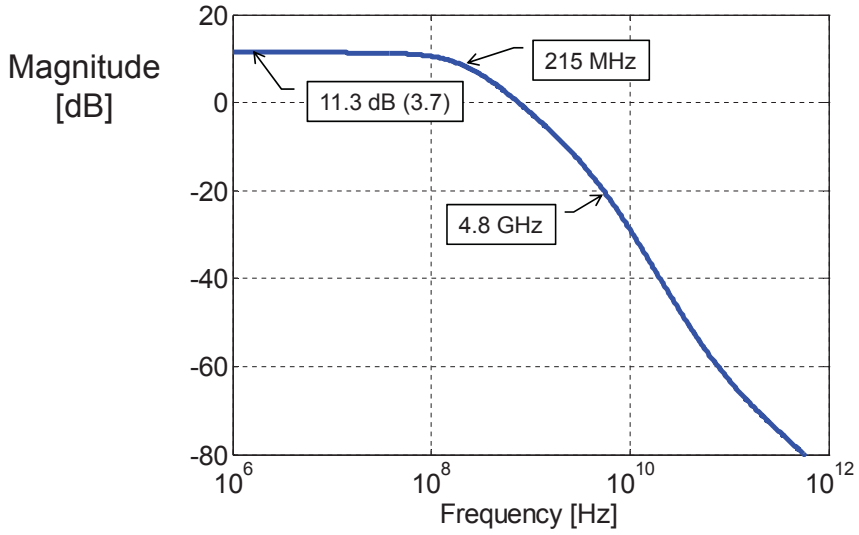
cbb $\equiv C_{gb} + C_{sb} + C_{db}$

cgs $\equiv C_{gs}$

cgd $\equiv C_{gd}$

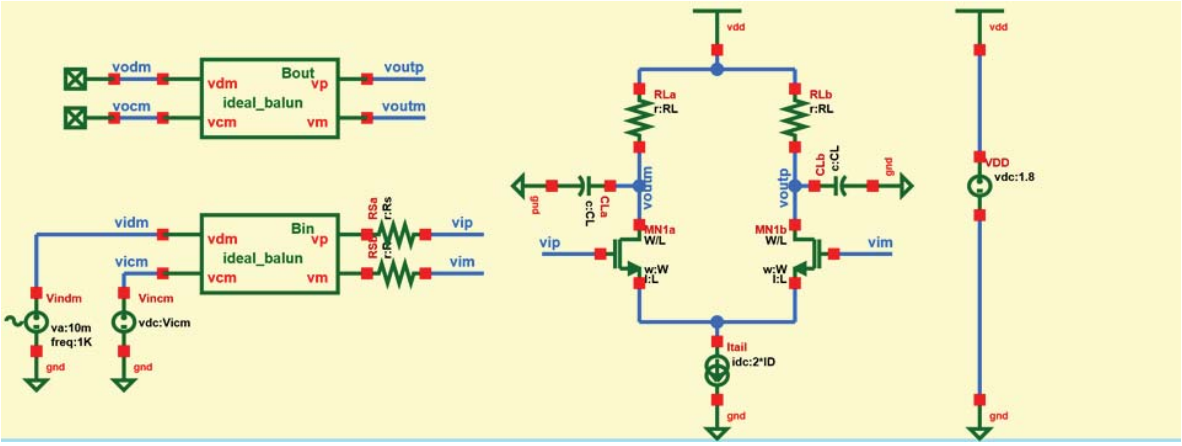
See Cadence help/manuals for more parameters and details: 'cdnshelp' from the command prompt, then search for bsim3v3 and see the section "Operating-Point Parameters"

Simulated AC Response



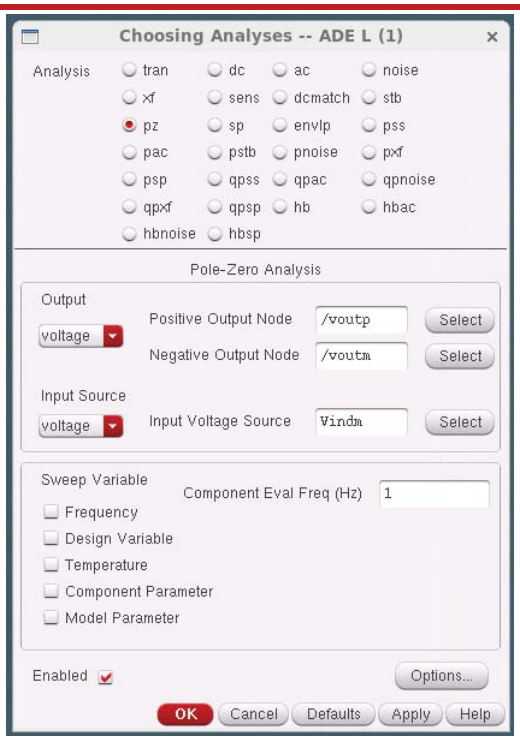
- Calculated values: $|A_{v0}|$=12 dB (4.0), f_{p1} = 200 MHz, f_{p2}= 5.8 GHz

Cadence Schematic

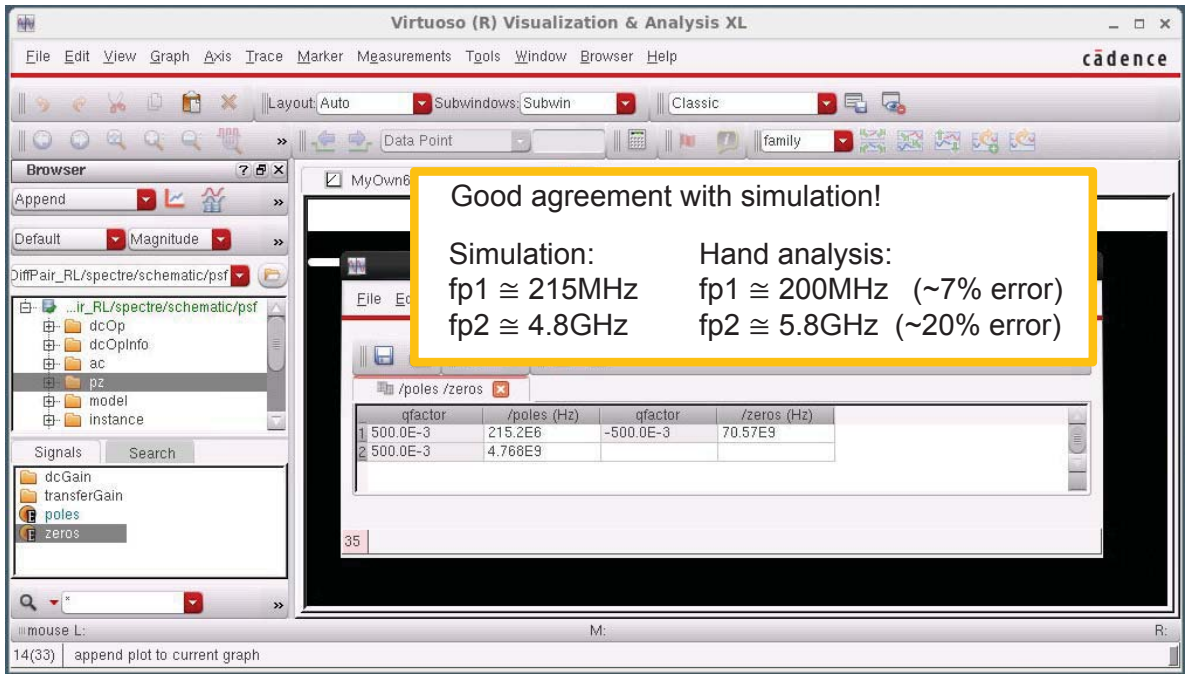


- W, L, Rs, RL, CL, ID are setup as design variables in ADE L
- Ideal baluns (from the ECE6720 library) are used to handle differential and common mode conversion

Using Pole/Zero Analysis (pz)



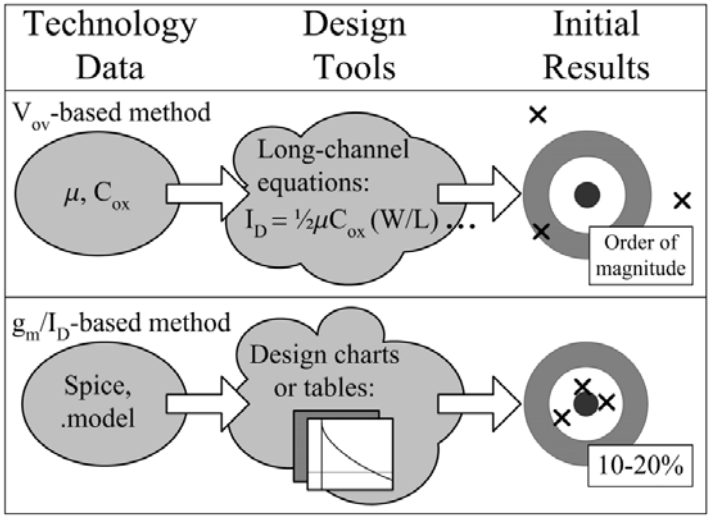
pz Analysis Results



Observations

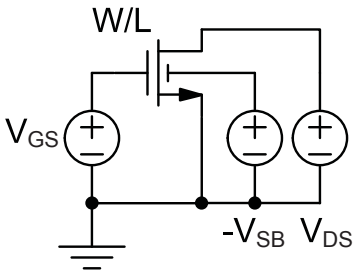
- The design is essentially right on target!
  - Typical discrepancies are no more than 10-20%, due to V_{DS} dependencies, finite output resistance, etc.
- We accomplished this by using pre-computed SPICE data in the design process
- Even if discrepancies are more significant, there's always the possibility 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



Advanced Characterization

- While variations due to V_{DS} are fairly small for most parameters, we can get better accuracy by sweeping V_{DS} as well
  - Adds another dimension to the tables
    - Harder to visualize
    - Easy to deal with using Matlab lookup functions instead of charts
- In many circuits $V_S \neq V_B$
  - e.g. a differential pair
  - Back-gate effect!
    - Sweeping V_{SB} generates characterization data that captures this effect as well



Simulation Data in Matlab

```
% data stored in /home/ece6720/matlab
>> 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]
     INFO: 'Univ of Utah ECE6720 models,
180nm CMOS, BSIM3'
      VGS: [73x1 double]
      VDS: [73x1 double]
      VSB: [11x1 double]
        L: [32x1 double]
        W: 5.0000e-06
    NFING: 1
>> size(nch.ID)
ans =
    32    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 in /home/ece6720/matlab

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

ans =
   8.4181e-06

>> help lookup
  Rev. 20140731, Boris Murmann
  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 modes (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;
```

Matlab Design Script for the Diff Pair 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
```

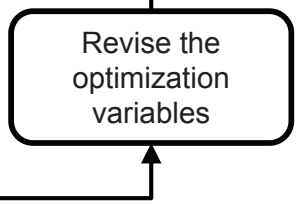

lookup_examples.m

```
% Basic usage examples for function "lookup"
% Boris Murmann
% Stanford University
% Rev. 20140731
% Modified by Ross Walker, Univ of Utah, 8/1/2014
clear all; close all;

%Plot settings
font_size = 20; font_name = 'Arial'; linewidth = 1;

% Load data table
load 180n.mat;

% Plot drain characteristics for different VGS at minimum L (default)
vgs = 0:0.2:max(nch.VGS);
id = lookup(nch, 'ID', 'VGS', vgs, 'VDS', nch.VDS);
figure(1)
set(gca,'FontSize',font_size,'FontName',font_name);
plot(nch.VDS, id','LineWidth',linewidth)
xlabel('V_D_S [V]')
ylabel('I_D [A]')

% Plot fT against gm_ID for different L
gm_id = 5:0.1:20;
wt = lookup(nch, 'GM_CGG', 'GM_ID', gm_id, 'L', nch.L);
figure(2)
```

An Example Optimization Flow

- Complicated circuits have *many* degrees of freedom and objectives
  - Usually we must make some heuristic choices up front
  - Charts and lookup tables help you iterate through possible designs rapidly

1) Pick transistor lengths
2) Pick g_m/I_D bias points
3) Determine transconductances
4) Determine bias currents (from g_m and g_m/I_D)
5) Total power consumption = ???
   - Room for improvement?
   - Ready to verify?
6) Determine W (from I_D/W)
7) Simulate the circuit: meets specs and estimated performance?
   - If yes, then you're done!
   - If not, revise the design flow: correct mistakes, improve estimates, etc.



A Note on Tools

- Modern IC design depends on *many* tools
  - Schematic editors, simulators, layout editors, verification tools, etc…
  - 'Hand' analysis tools
    - MATLAB, Octave, MathCAD, Maple, Excel, etc…
    - Technology characterization charts and lookup tables
  - GUIs & graphics versus scripting

- The designer **uses the tools** to design the circuit
  - Tools don't do the design for you
  - Tools sometimes break, or are incompatible with each other
    - Search/read the manuals, learn to use tools correctly
      - Tips: ctrl + f, google, cdnshelp, CAD tutorial

- Pick the right tool for the job
  - Balance complexity, overhead, connection to the design objectives
    - This can be a very personal choice
  - Sometimes biting the bullet is better than fancy solutions

Chapter 5 Summary

- Revisited the three figures of merit for transistors, in the context of ***design***
  - g_m/I_D, ω_T, g_m/g_{ds}
  - Device characteristics that are directly linked to circuit performance
    - Used as tradeoff 'knobs' or indices to balance power efficiency, speed, gain
    - Powerful parameters to characterize a process technology
- Saw how the square law breaks down in practice
  - Weak/moderate/strong inversion, short channel effects
- Discussed g_m/I_D-based design using charts and lookup tables
  - Provides a systematic design/optimization methodology
    - Accurately estimate capacitances, gain, power dissipation
    - Minimize tweaking of W and L
- Worked through a differential pair design example
  - Good agreement between hand analysis and simulation
  - This was a basic design exercise: nothing to optimize

Chapter 5 Learning Objectives

- Learning objectives
  - Understand tradeoffs between speed, power efficiency, and gain
    - In transistors, and in circuits
  - Be able to use technology characterization tables/charts to size transistors
    - Get practice in the design project
  - Develop a systematic design methodology in the absence of simple analytical models like the square law equation

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 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 g_m/I_D Lookup Table Methodology,” IEICE Trans. Electronics, Vol. E94-C, No.3, Mar. 2011.