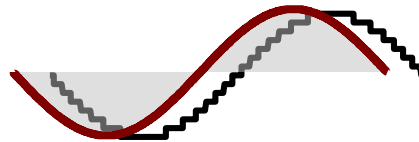

Systematic Design of Analog Circuits Using Pre-Computed Lookup Tables

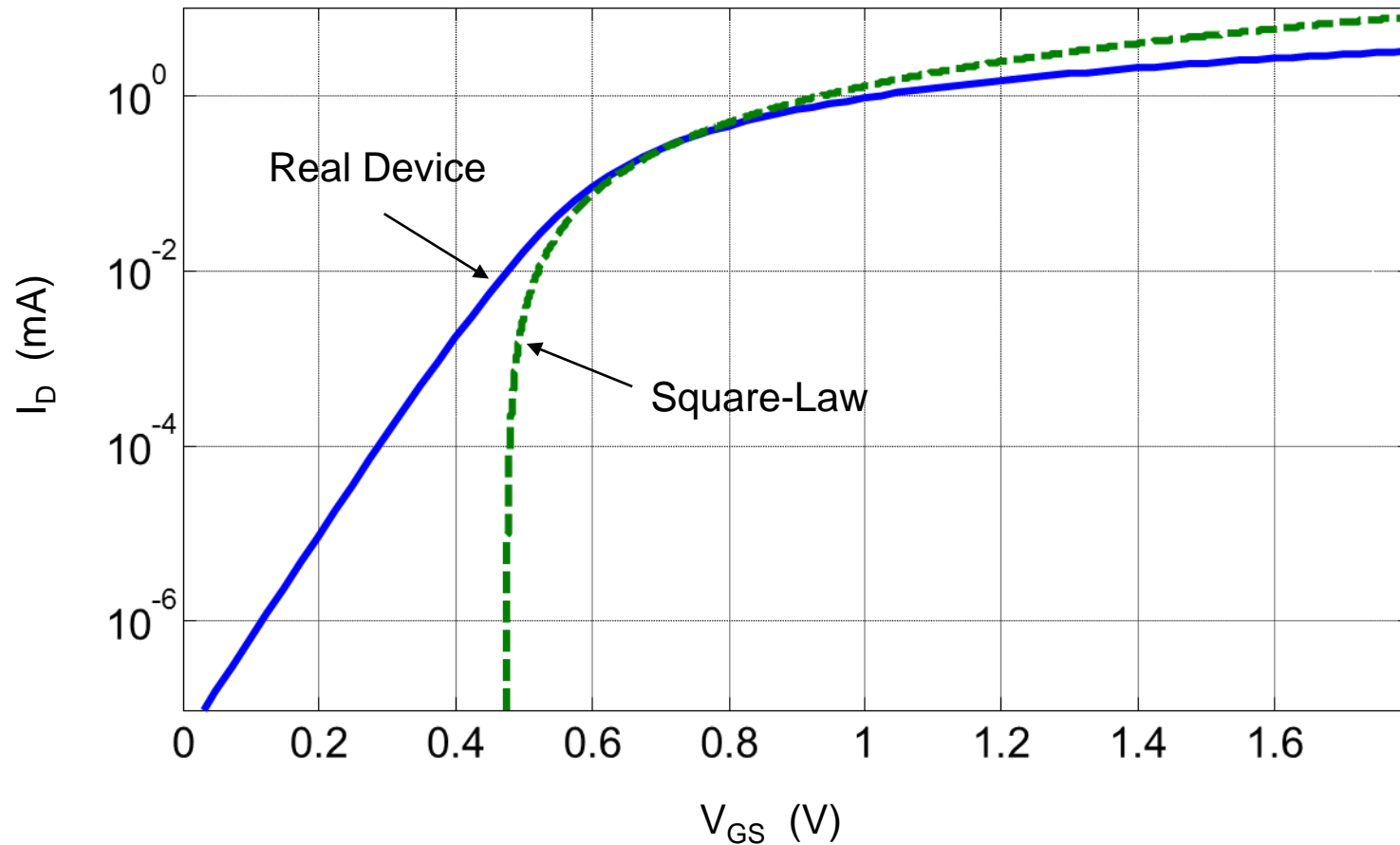


February 26, 2016
Boris Murmann
Stanford University
murmann@stanford.edu

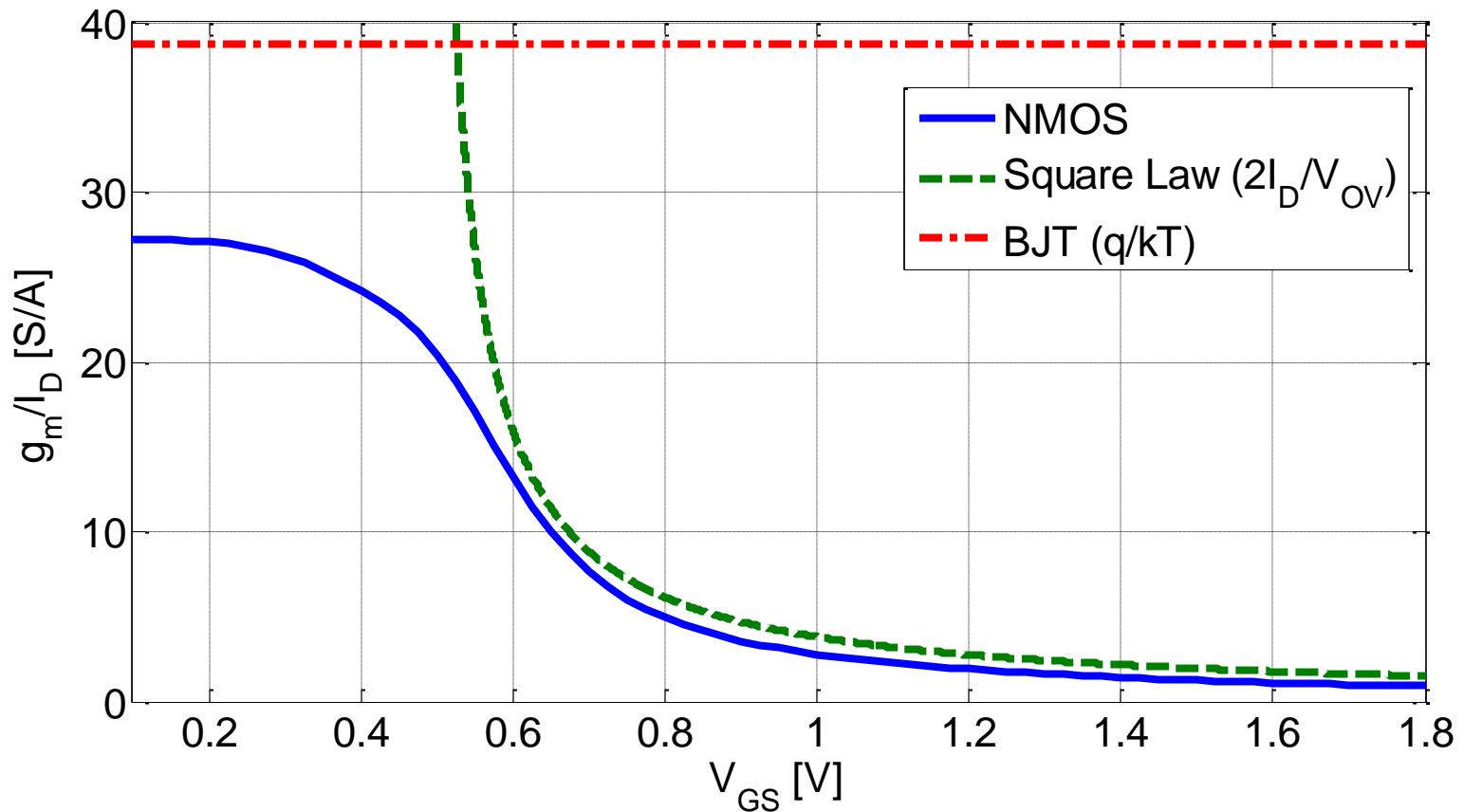
Outline

- Motivation
- Examples
 - Simple differential pair
 - Noise-limited gain stages
 - Distortion-limited gain stages
 - Design with process corners

Motivation: Complex MOS I-V

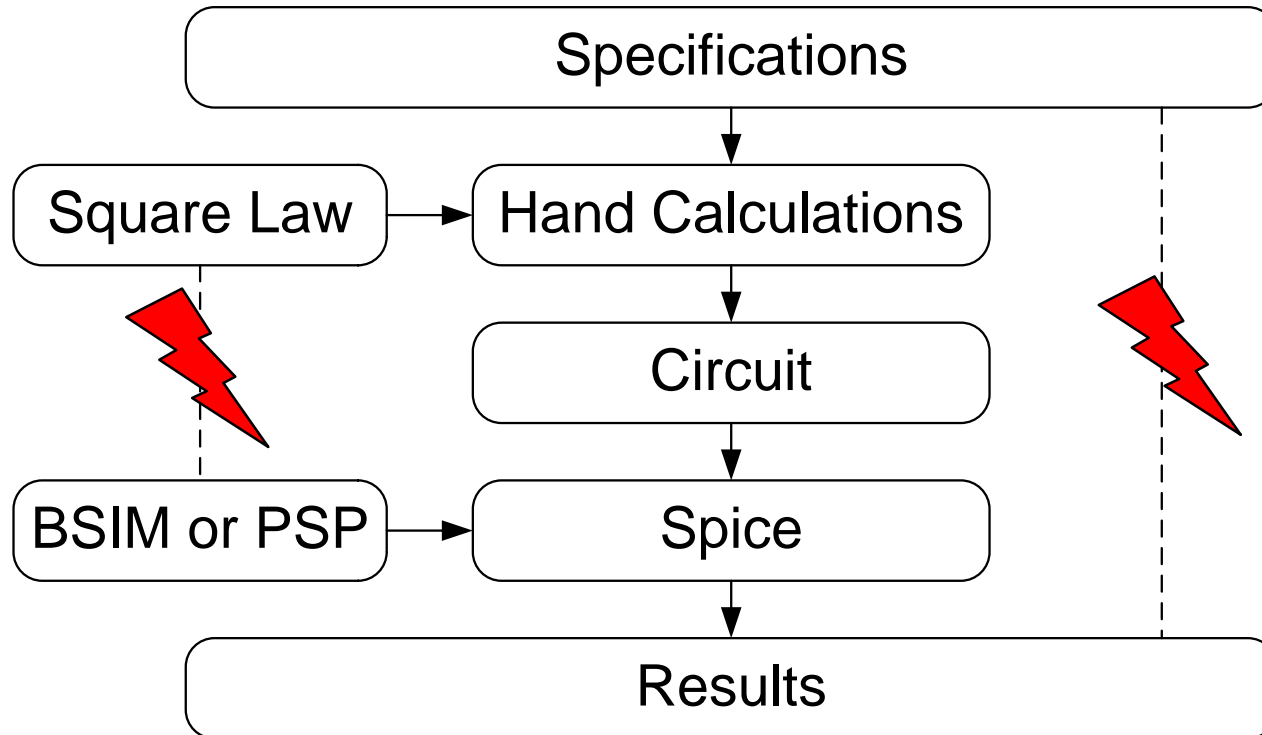


$$g_m/I_D$$



- The square law fails miserably at predicting g_m/I_D in moderate and weak inversion

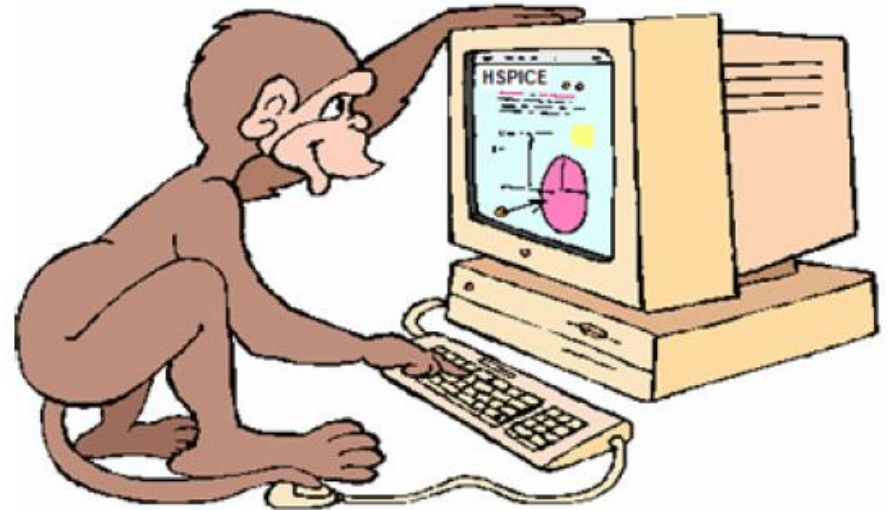
The Problem



- 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

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



Starting Point: Technology Characterization via DC Sweep

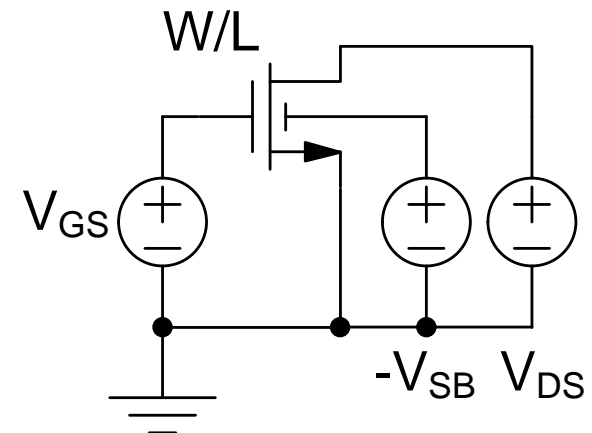
```
* /usr/class/ee214b/hspice/techchar.sp

.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'
vgasn     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_cggs  = par('cggsbo(mn)')
.probe n_cggs  = 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

```
% 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]
    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;
```

Download

<http://web.stanford.edu/~murmann/gmid>

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 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}

$$\frac{g_m}{C_{gg}}$$

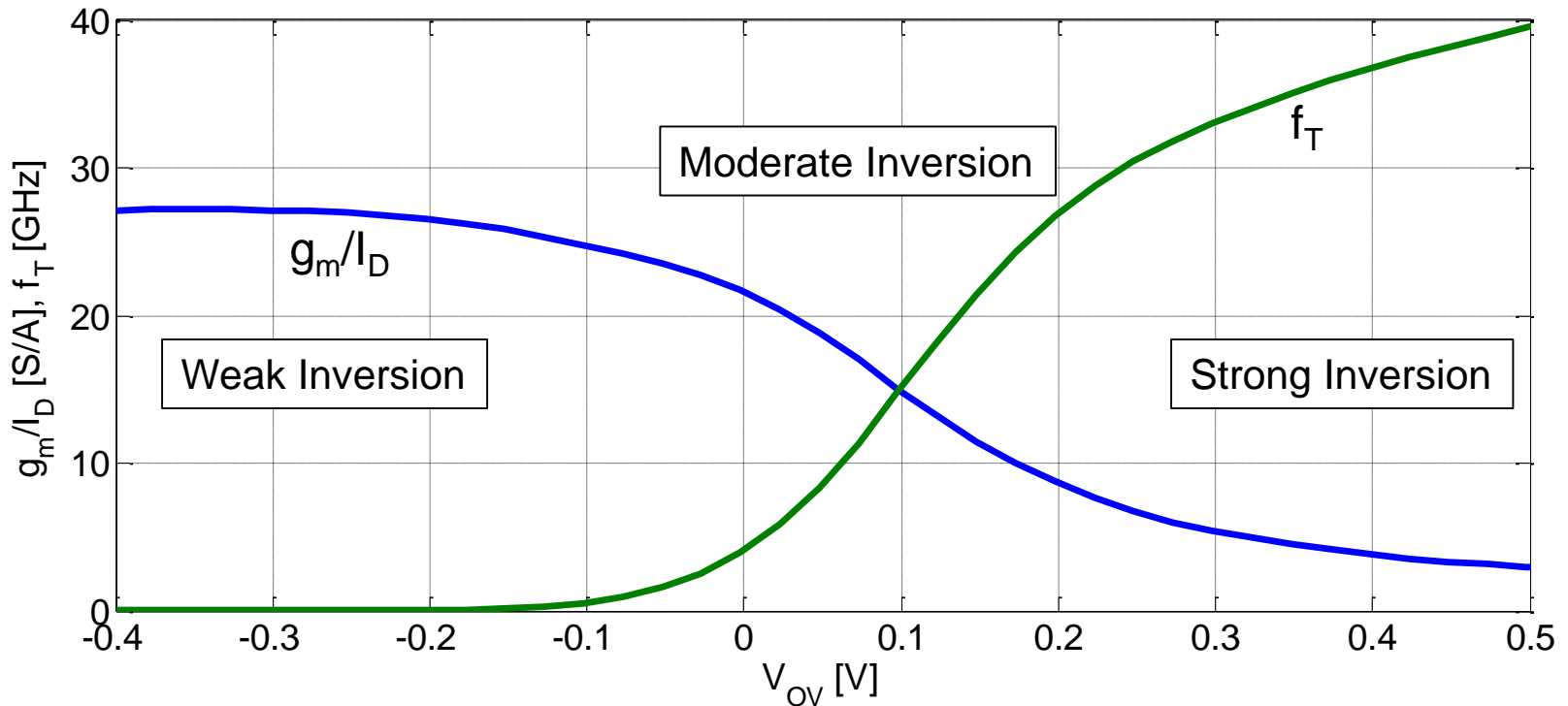
$$\approx \frac{3}{2} \frac{\mu V_{OV}}{L^2}$$

- Intrinsic gain
 - Want large g_m , but no g_o

$$\frac{g_m}{g_o}$$

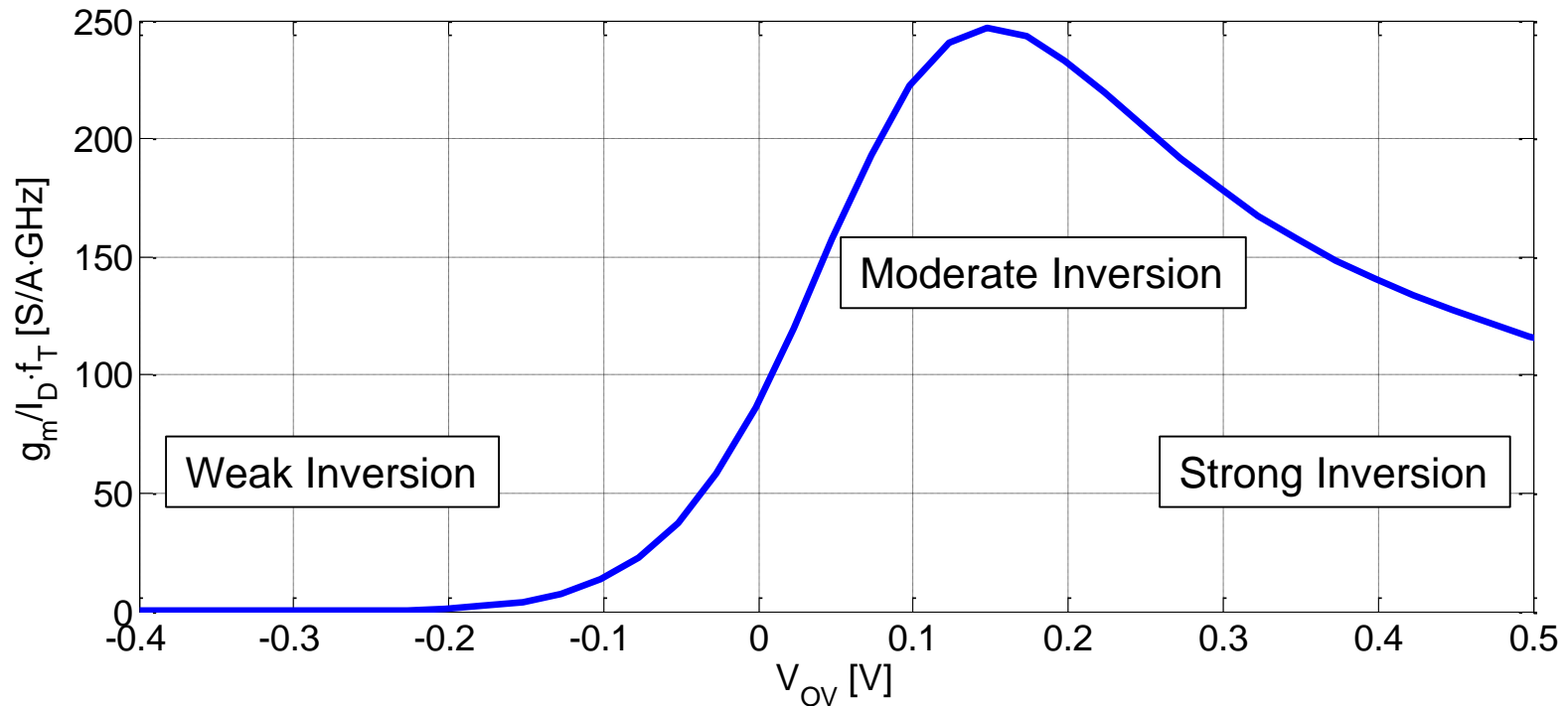
$$\approx \frac{2}{\lambda V_{OV}}$$

Design Tradeoff: g_m/I_D and f_T



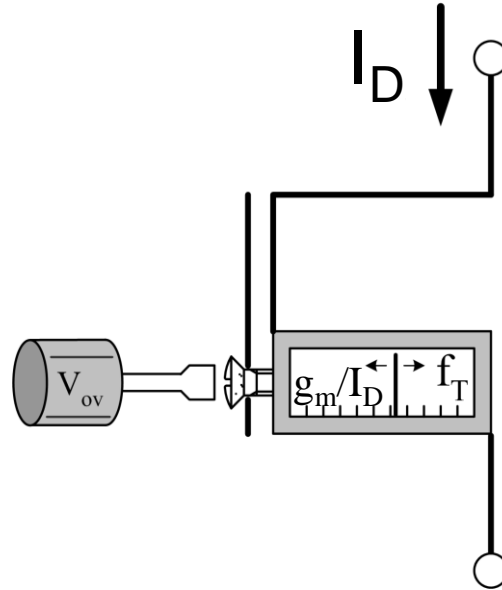
- Weak inversion: Large g_m/I_D (>20 S/A), but small f_T
- Strong inversion: Small g_m/I_D (<10 S/A), but large f_T

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)

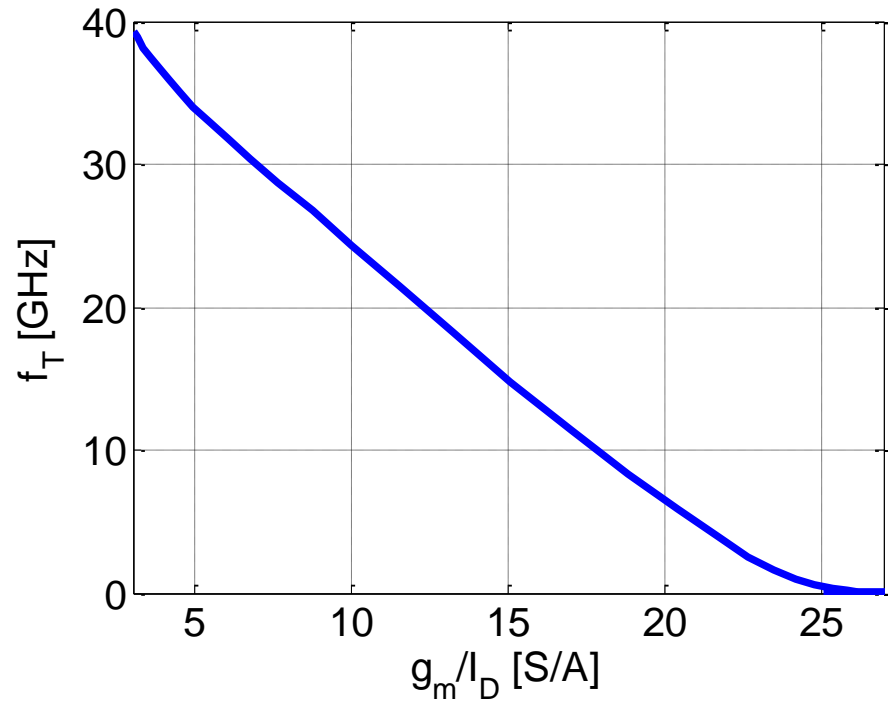
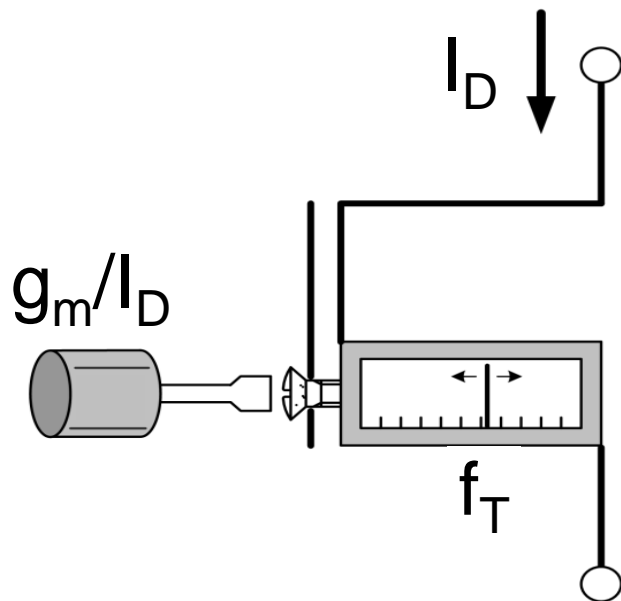
Design in a Nutshell



- 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 $2 / (g_m/I_D)$

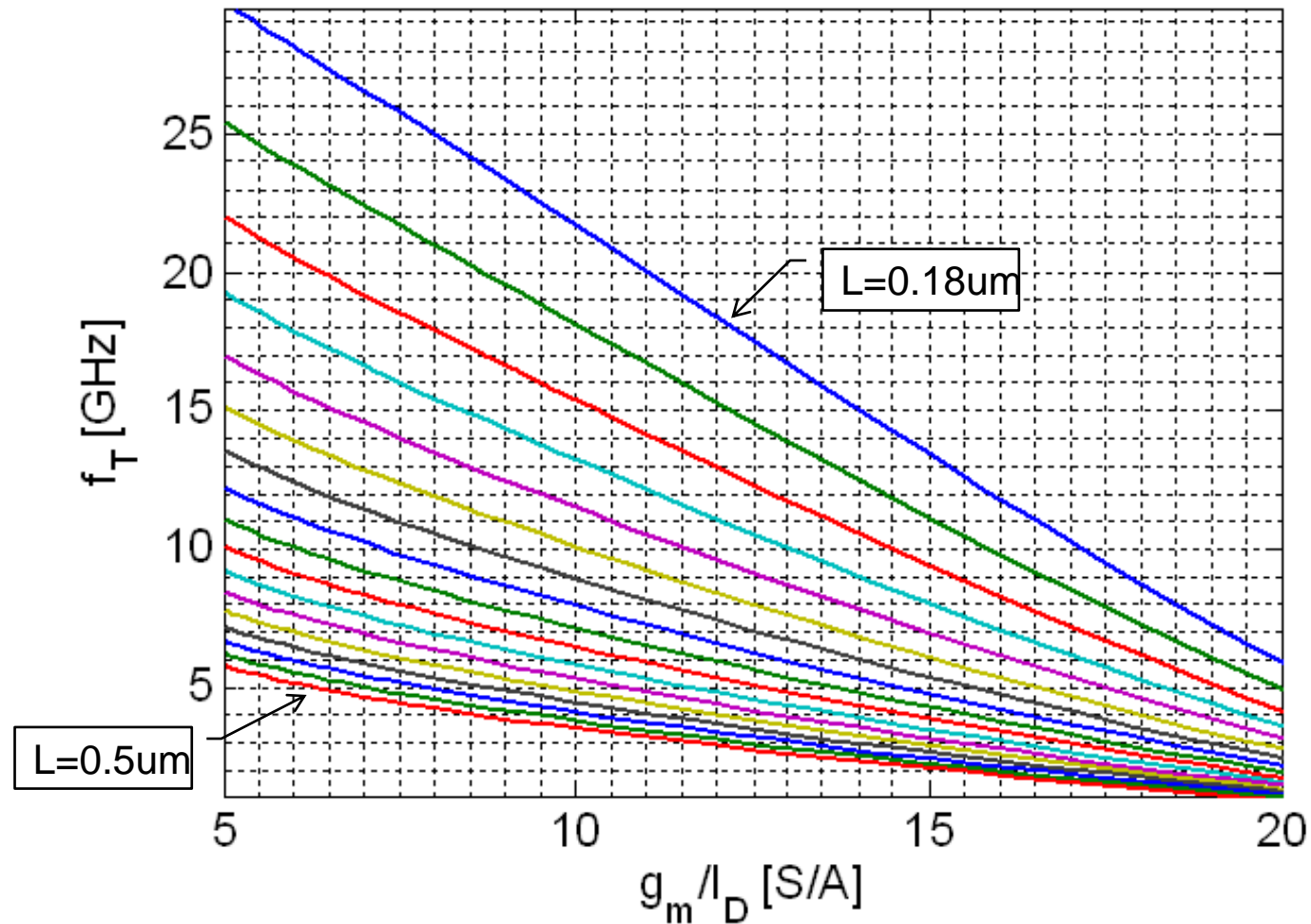


g_m/I_D -centric Technology Characterization

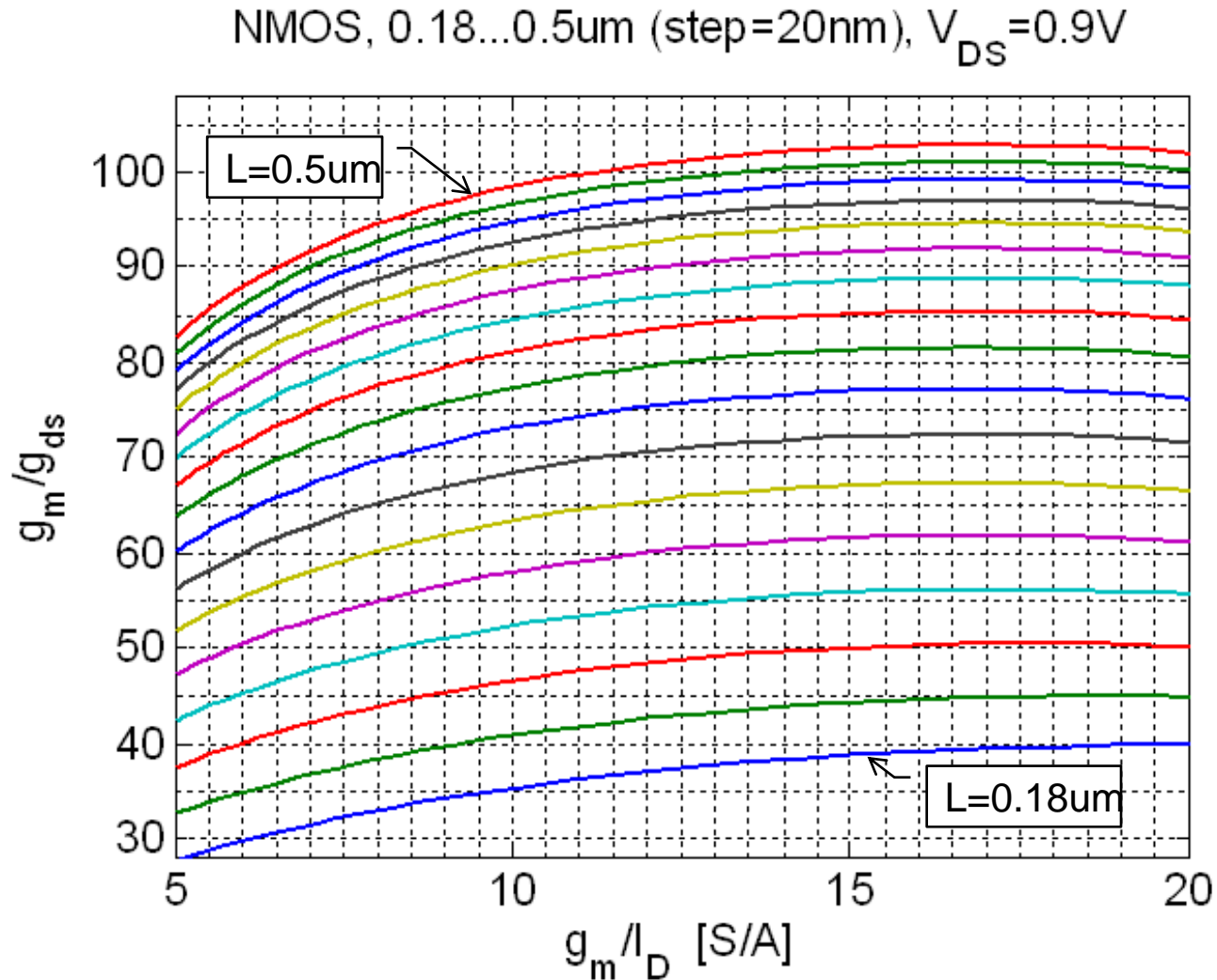
- Plot the following parameters for a reasonable range of g_m/I_D and channel lengths
 - Transit frequency (f_T)
 - Intrinsic gain (g_m/g_o)
- Can also plot relative estimates of extrinsic capacitances
 - C_{gd}/C_{gg} and C_{dd}/C_{gg}
- Note that all of these parameters are (to first order) independent 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

NMOS, 0.18...0.5 μ m (step=20nm), $V_{DS}=0.9V$

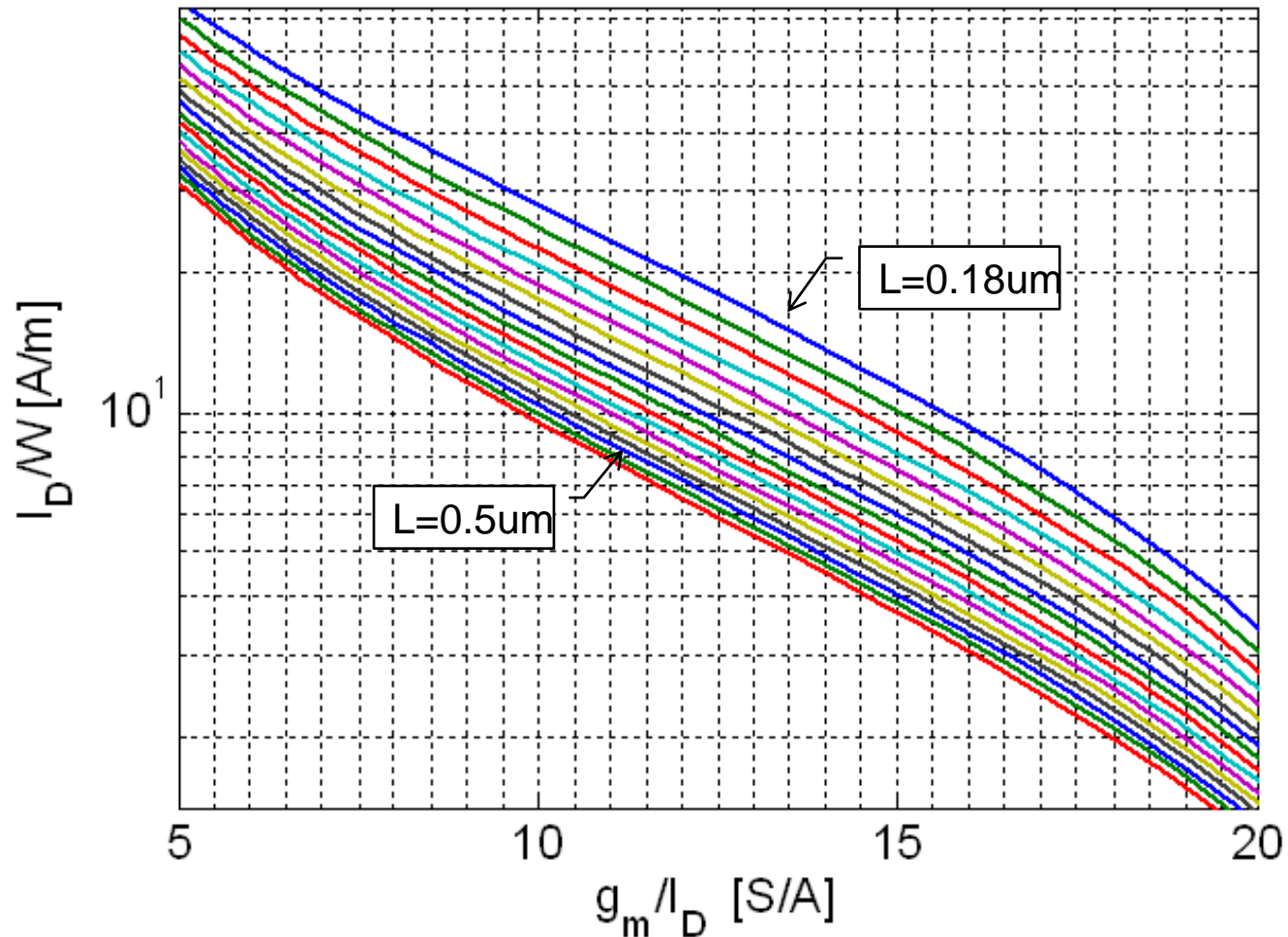


Intrinsic Gain Chart

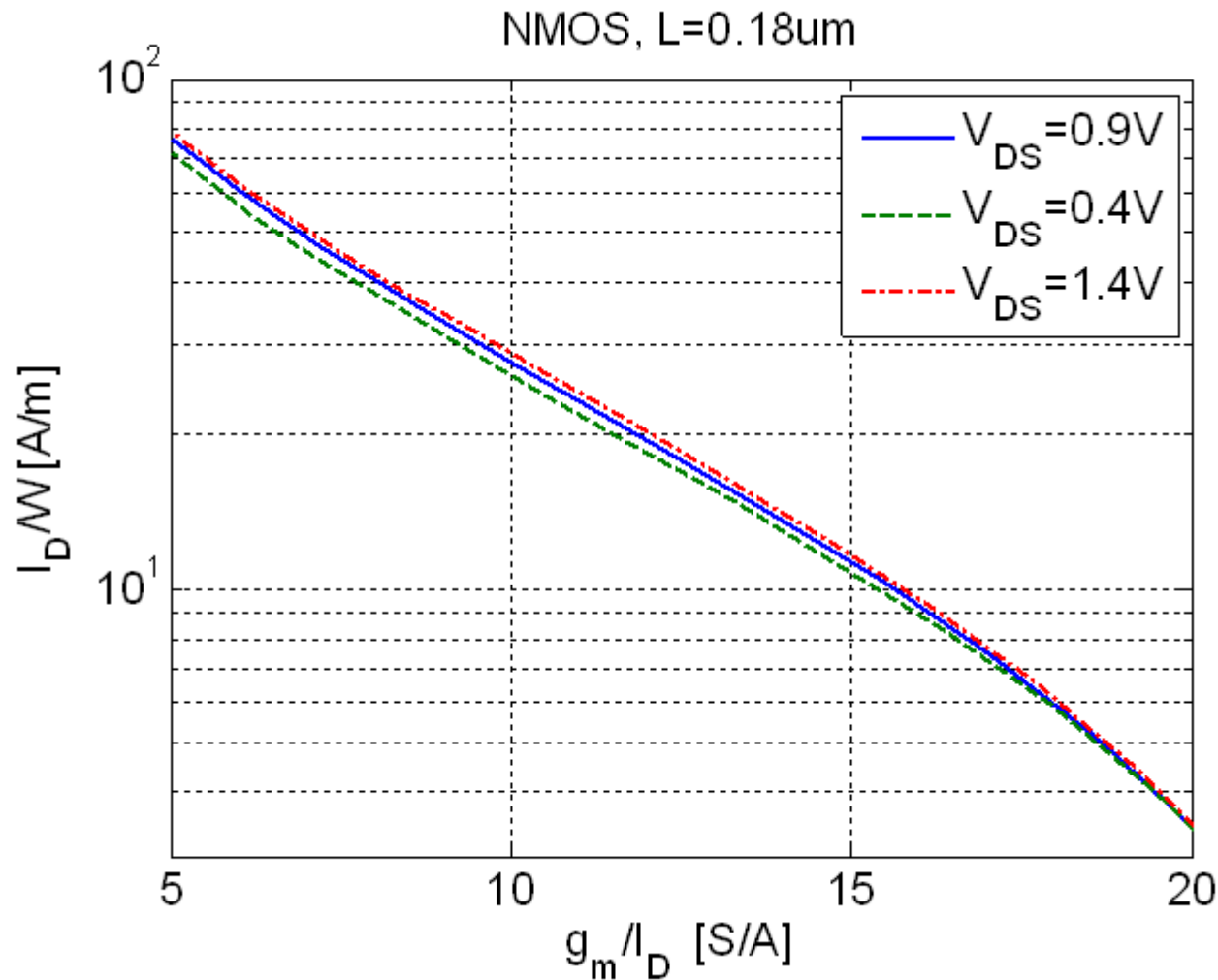


Current Density Chart

NMOS, 0.18...0.5 μm (step=20nm), $V_{\text{DS}}=0.9\text{V}$



V_{DS} Dependence



- V_{DS} dependence is relatively weak
- Typically OK to work with data generated for $V_{DD}/2$
- But, no problem to use the actual V_{DS} (or a good estimate of it) in the lookup command

Generic Design Flow

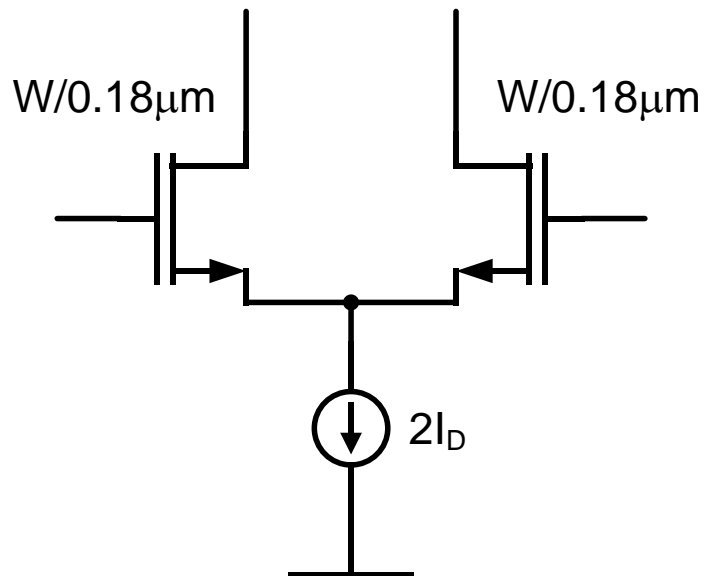
- 1) Determine g_m (from design objectives)
- 2) Pick L
 - Short channel \rightarrow high f_T (high speed)
 - Long channel \rightarrow high intrinsic gain
- 3) Pick g_m/I_D (or f_T)
 - Large $g_m/I_D \rightarrow$ low power, large signal swing (low V_{Dsat})
 - Small $g_m/I_D \rightarrow$ high f_T (high 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)

Outline

- Motivation
- Examples
 - Basic differential pair
 - Noise-limited gain stages
 - Distortion-limited gain stages
 - Design with process corners

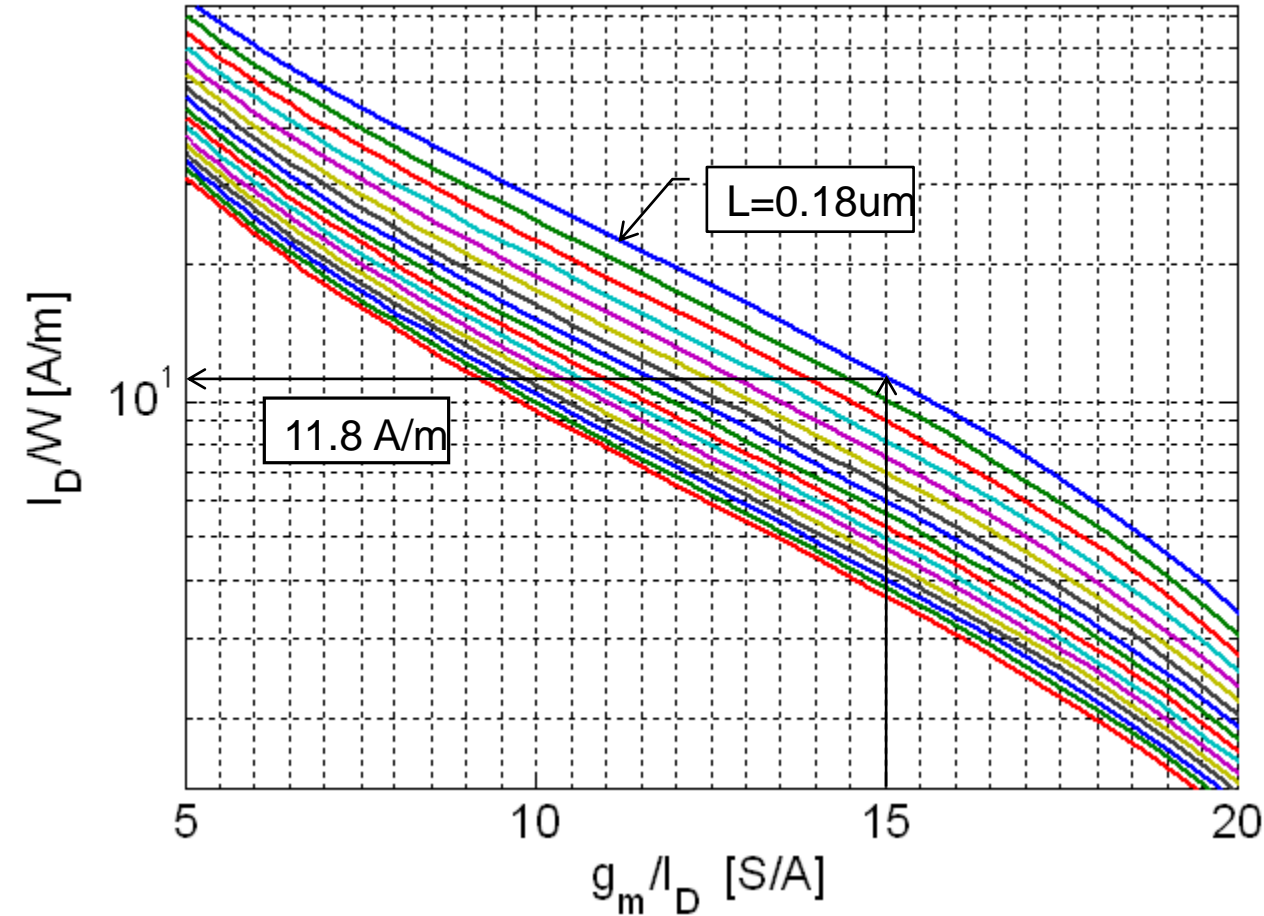
Basic Sizing Example



- Determine the width of the differential pair transistors such that $g_m = 10\text{mS}$
- Consider various levels of inversion (weak, moderate, strong)
- For each case, compute the required I_D , W and C_{gg} (total gate capacitance)

Current Density Lookup

NMOS, 0.18...0.5um (step=20nm), $V_{DS}=0.9V$

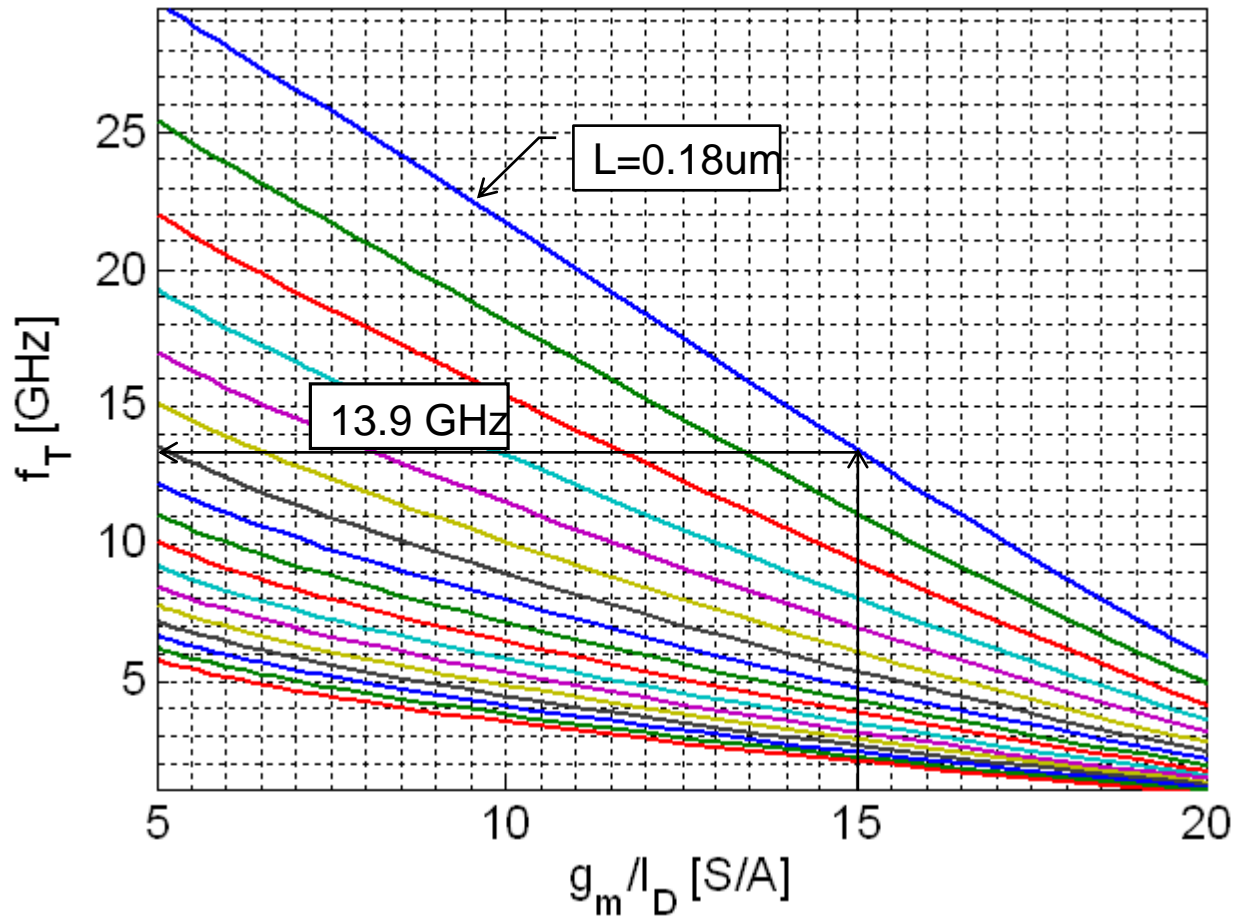


$$I_D = \frac{g_m}{\frac{g_m}{I_D}} = \frac{10mS}{15\frac{S}{A}} = 0.67mA$$

$$W = \frac{I_D}{\frac{I_D}{W}} = \frac{0.67mA}{11.8\frac{A}{m}} = 56.6\mu m$$

Determine C_{gg} via f_T Look-up

NMOS, 0.18...0.5 μ m (step=20nm), $V_{DS}=0.9V$



$$C_{gg} = \frac{g_m}{\omega_T}$$

$$C_{gg} = \frac{10mS}{2\pi \cdot 13.9GHz} = 114fF$$

Matlab Script

```
% Basic sizing example
clear all;
close all;

% Load parameters
load 180nch.mat;

% Specification
gm = 10e-3;

% Chosen inversion levels and resulting drain current
gm_id = [25 15 5]';
ID = gm./gm_id

% Current density and width
JD = lookup(nch, 'ID_W', 'GM_ID', gm_id)
W = ID./JD

% Transit frequency and Cgg
wT = lookup(nch, 'GM_CGG', 'GM_ID', gm_id);
Cgg = gm./wT
```

Result

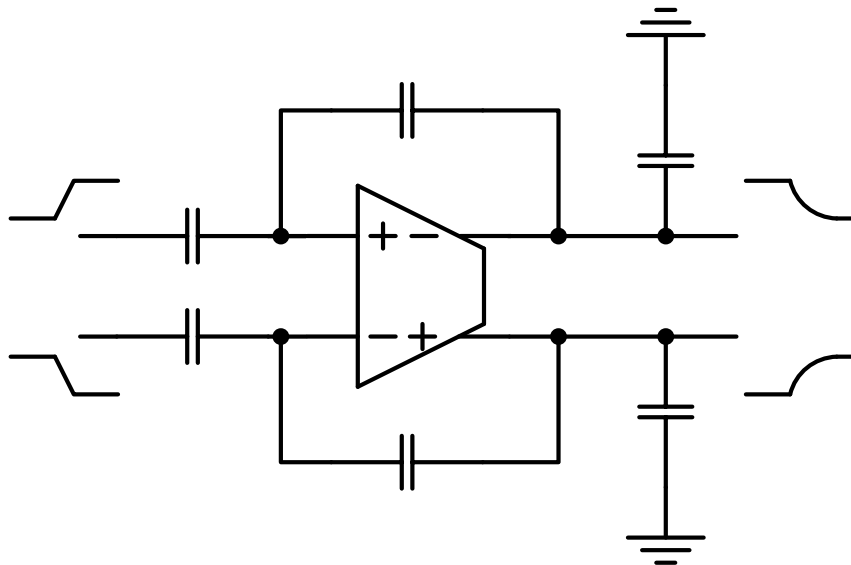
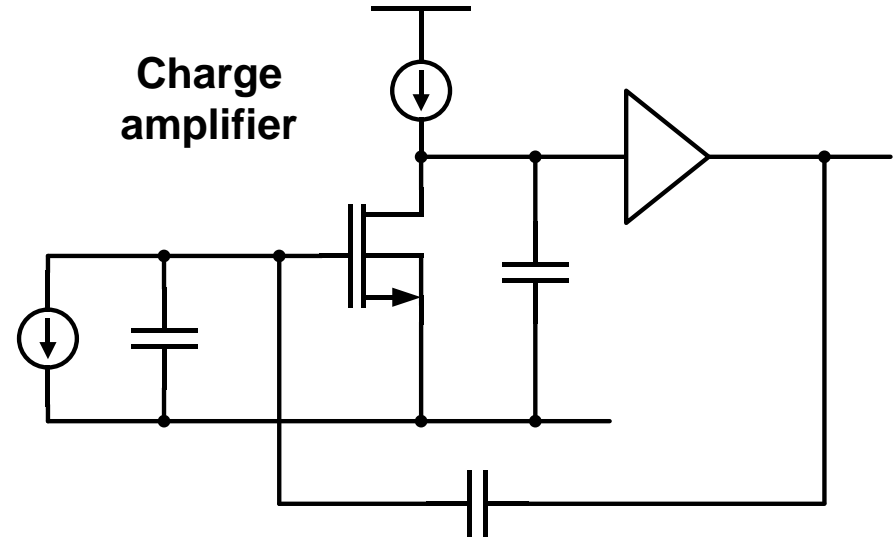
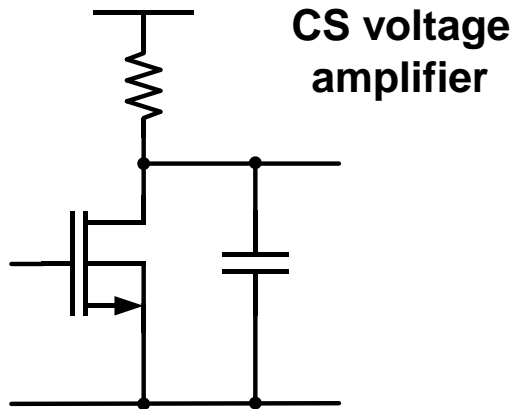
g_m/I_D [S/A]	25 (weak inversion)	15 (moderate inversion)	5 (strong inversion)
g_m [mS]	10	10	10
I_D [mA]	0.4	0.67	2
I_D/W [A/m]	0.12	11.8	83.3
W [μm]	3243	56.7	24.0
f_T [GHz]	0.37	13.9	32.3
C_{gg} [fF]	4346	114	49

- Weak inversion
 - Small current, large device, large capacitance
- Strong inversion
 - Large current, small device, small capacitance
- Moderate inversion
 - A good compromise!

More Examples

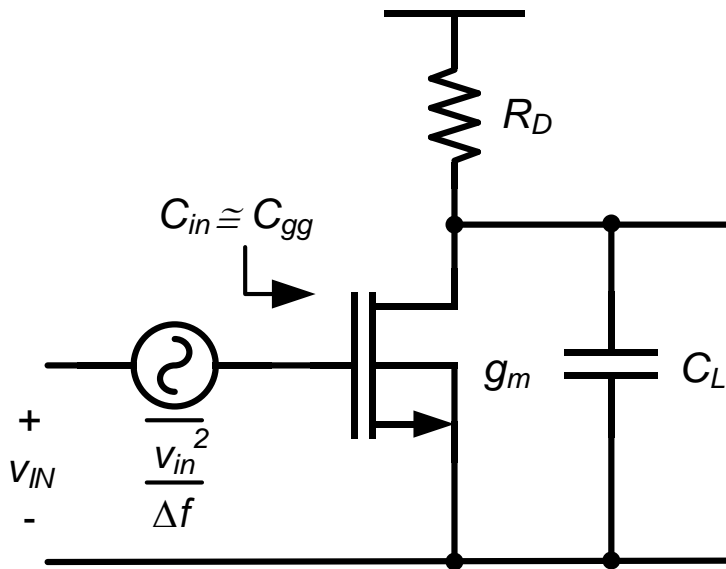
- Noise-limited gain stages
 - CS voltage amplifier
 - Basic charge amplifier
 - Basic OTA for switched-capacitor circuits
 - Folded-cascode OTA for switched-capacitor circuits
- Distortion-limited gain stages
- Design with process corners

Noise-Limited Gain Stages — Overview



SC amplifier
→ Basic OTA
→ Folded cascode OTA

CS Voltage Amplifier



Input-referred noise

$$\frac{\overline{v_{in}^2}}{\Delta f} \cong 4kT\gamma_n \frac{1}{g_m}$$

Gain-bandwidth product

$$GBW = \frac{1}{2\pi} \frac{g_m}{C_L} = \frac{1}{2\pi} \frac{g_m}{FO \cdot C_{gg}} = \frac{f_T}{FO}$$

“Fan-out”

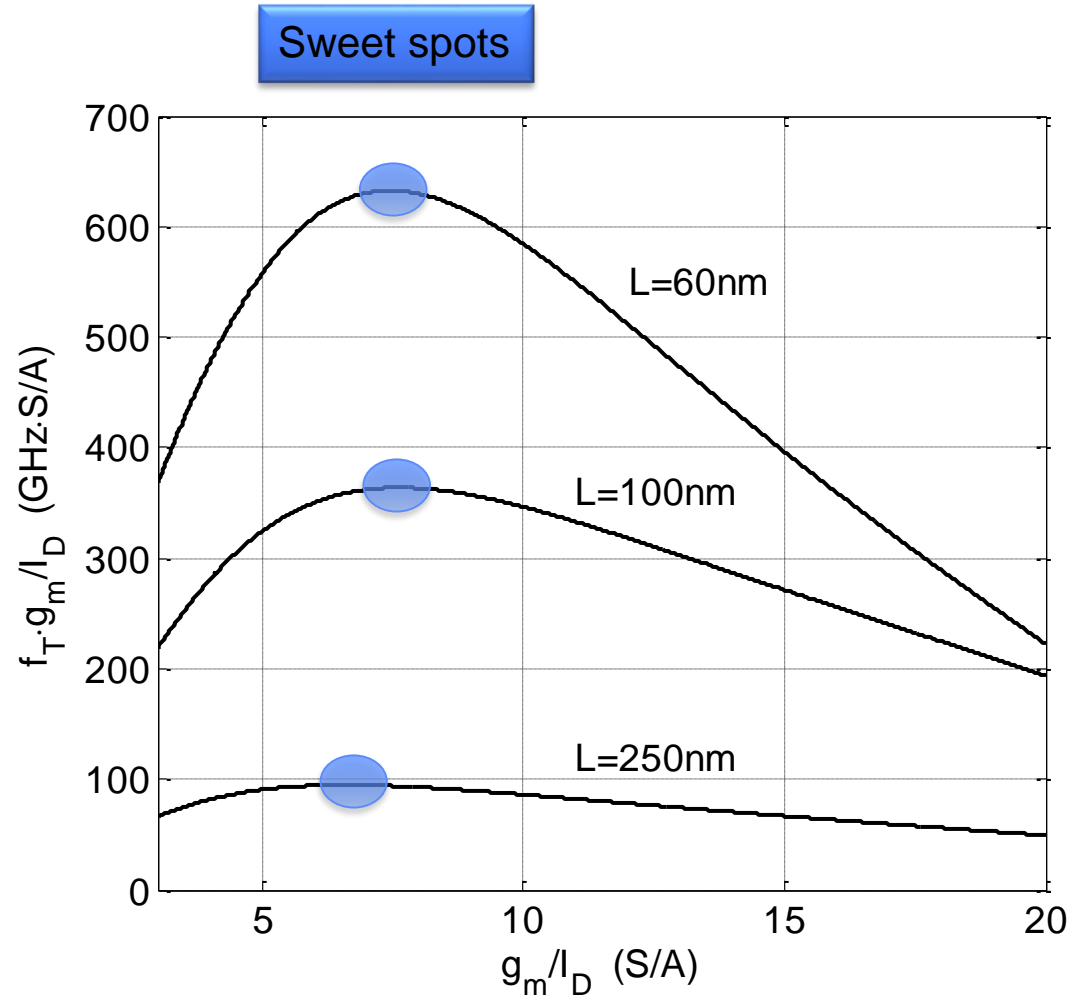
$$FO = \frac{C_L}{C_{in}} \cong \frac{C_L}{C_{gg}}$$

Efficiency

$$\frac{GBW}{I_D} = \frac{GBW}{g_m \cdot \frac{I_D}{g_m}} = \frac{\frac{f_T}{FO}}{\frac{4kT\gamma_n}{v_{in}^2} \cdot \frac{I_D}{g_m}} \propto \frac{g_m}{I_D} \cdot f_T$$

- For a given noise level and fan-out, the efficiency of the circuit scales with the product of g_m/I_D and f_T
- This has been recognized and used in the design of RF low-noise amplifiers, see e.g.
 - Shameli, ISLPD 2006
 - Taris, RFIC 2011

Product of g_m/I_D and f_T



Sizing Example

% Constants and design parameters

```
kB = 1.38e-23;  
T = nch.TEMP;  
gamma_n = 1;  
vin_noise = 1e-9;  
L = 0.06;
```

% Set inversion level to sweet spot

```
gm_ID = 7;
```

% Compute gm based on noise specification

```
gm = 4*kB*T*gamma_n/vin_noise^2
```

% Compute drain current and device width

```
ID = gm/gm_ID  
JD = lookup(nch, 'ID_W', 'GM_ID', gm_ID, 'L', L);  
W = ID/JD
```

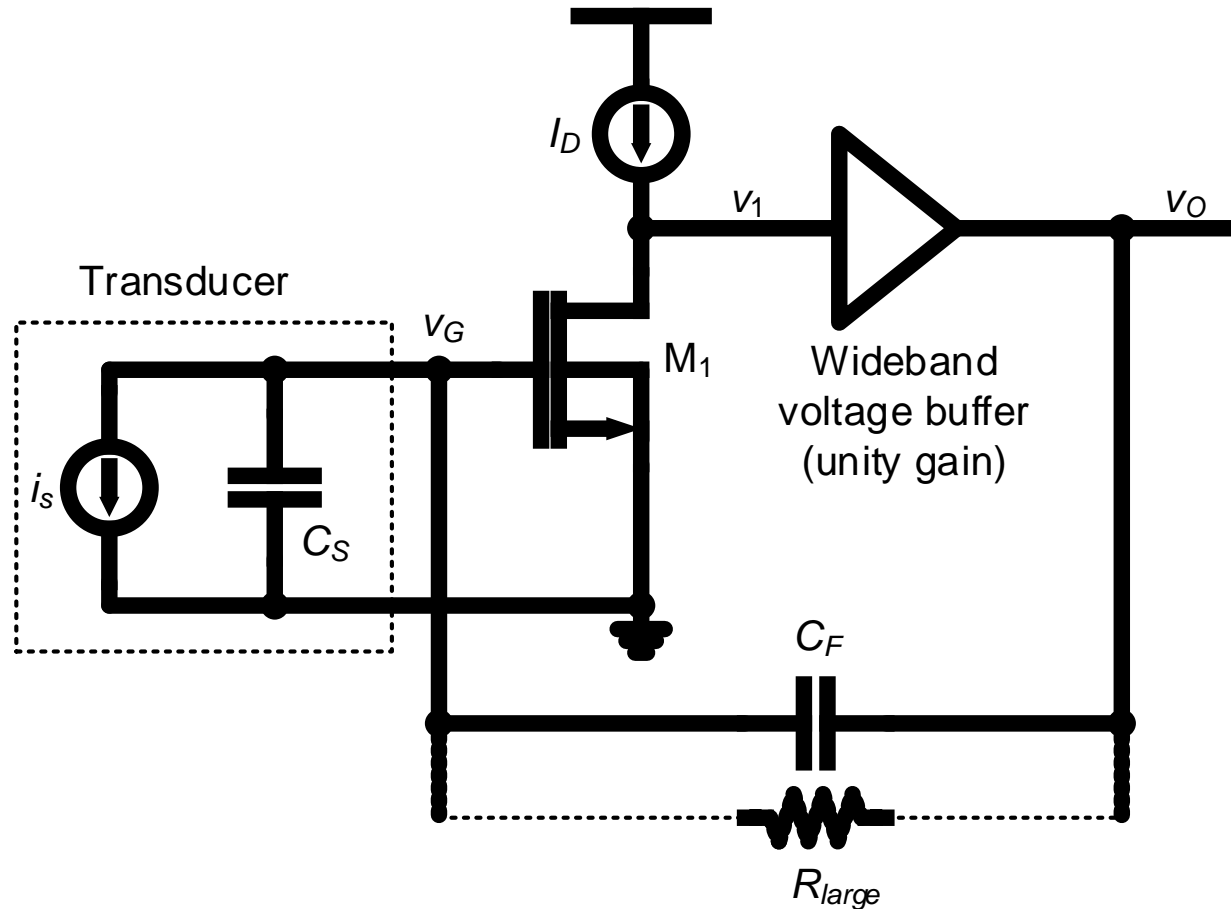
Results:

$g_m = 16.6 \text{ mS}$

$I_D = 2.4 \text{ mA}$

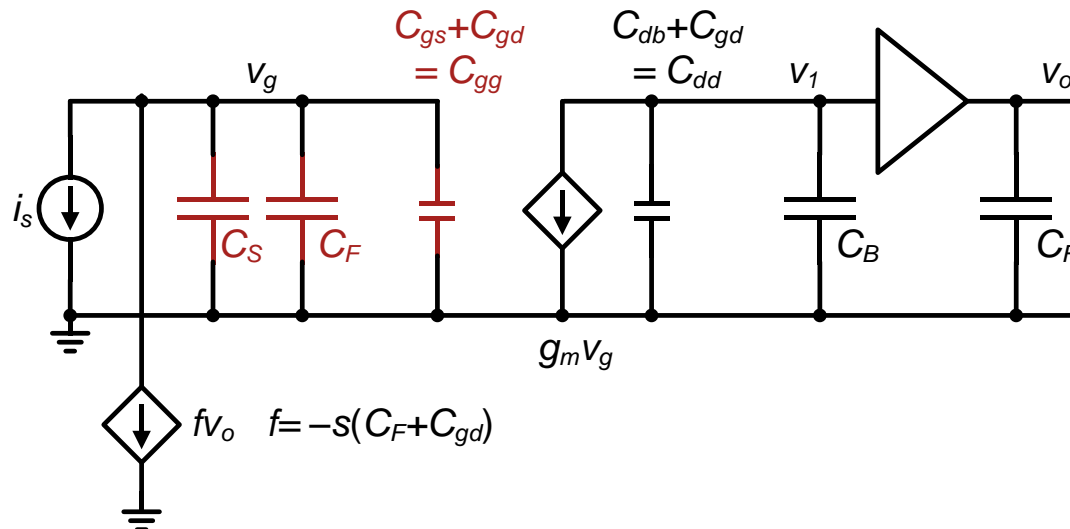
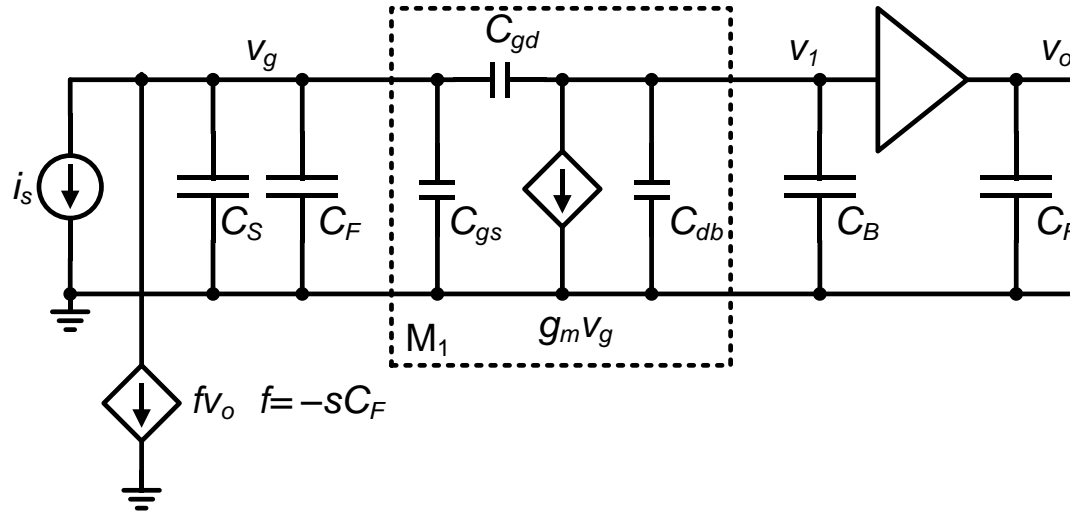
$W = 28 \text{ }\mu\text{m}$

Basic Charge Amplifier

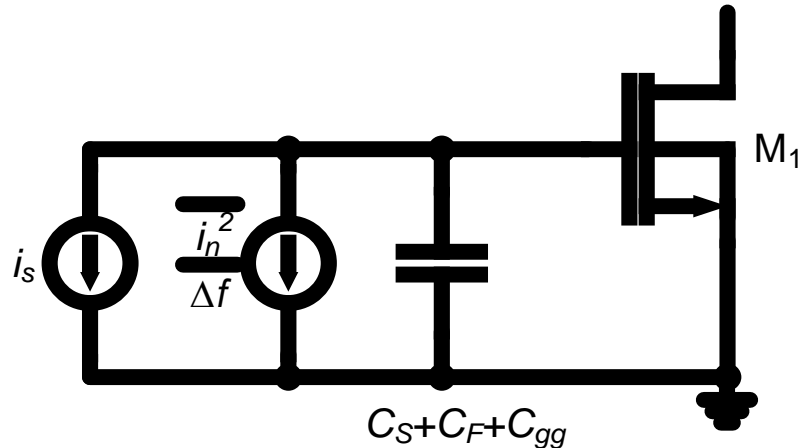
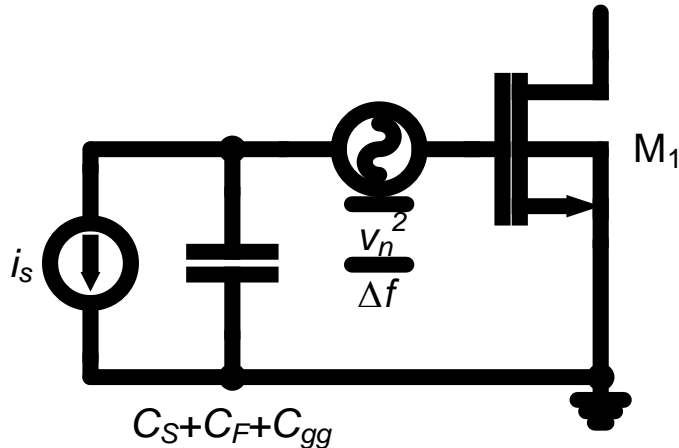


- Used in photodetectors, MEMS interfaces, etc.

Two-Port Model (Shunt-Shunt)



Relevant Sub-Circuit for Noise Analysis



$$\frac{\overline{i_n^2}}{\Delta f} = \frac{\overline{v_n^2}}{\Delta f} \omega^2 (C_S + C_F + C_{gg})^2 = \frac{4kT\gamma_n}{g_m} \omega^2 (C_S + C_F + C_{gg})^2$$

- Transconductance (g_m) and gate capacitance (C_{gg}) are fundamentally linked ($C_{gg}, g_m \propto W$)
- There exists an optimum device size

Scenario 1: Optimization Assuming Constant ω_T

$$g_m = \omega_T C_{gg} \qquad \frac{\overline{i_n^2}}{\Delta f} = \frac{4kT\gamma_n}{\omega_T \mathbf{C}_{gg}} \omega^2 (C_S + C_F + \mathbf{C}_{gg})^2$$

- The noise is minimized when the following term is minimized:

$$\frac{(C_S + C_F + \mathbf{C}_{gg})^2}{\mathbf{C}_{gg}} \qquad \boxed{\Rightarrow C_{gg,opt} = C_S + C_F}$$

- This is a classical textbook result
 - See e.g. Chan Carusone, 2011

Scenario 2: Optimization Assuming Square Law & Constant I_D

$$g_m = \sqrt{2I_D \mu C_{ox} \frac{W}{L}} \cong \sqrt{3I_D \frac{\mu}{L^2} C_{gg}}$$

$$\frac{\overline{i_n^2}}{\Delta f} = \frac{4kT\gamma_n}{\sqrt{3I_D \frac{\mu}{L^2} C_{gg}}} \omega^2 (C_S + C_F + C_{gg})^2$$

- The noise is minimized when the following term is minimized:

$$\frac{(C_S + C_F + C_{gg})^2}{\sqrt{C_{gg}}} \quad \Rightarrow \quad C_{gg,opt} = \frac{C_S + C_F}{3}$$

- This is another classical result
 - See e.g. Sansen, 1990

Issue & Solution

- The square does not apply to modern devices
- But, the optimum can be easily found using lookup table data

% Parameters

```
Cf_plus_Cs = 1e-12;
```

```
ID = 1e-3;
```

```
W = 5:1000;
```

```
L = [0.06 0.1 0.2 0.4];
```

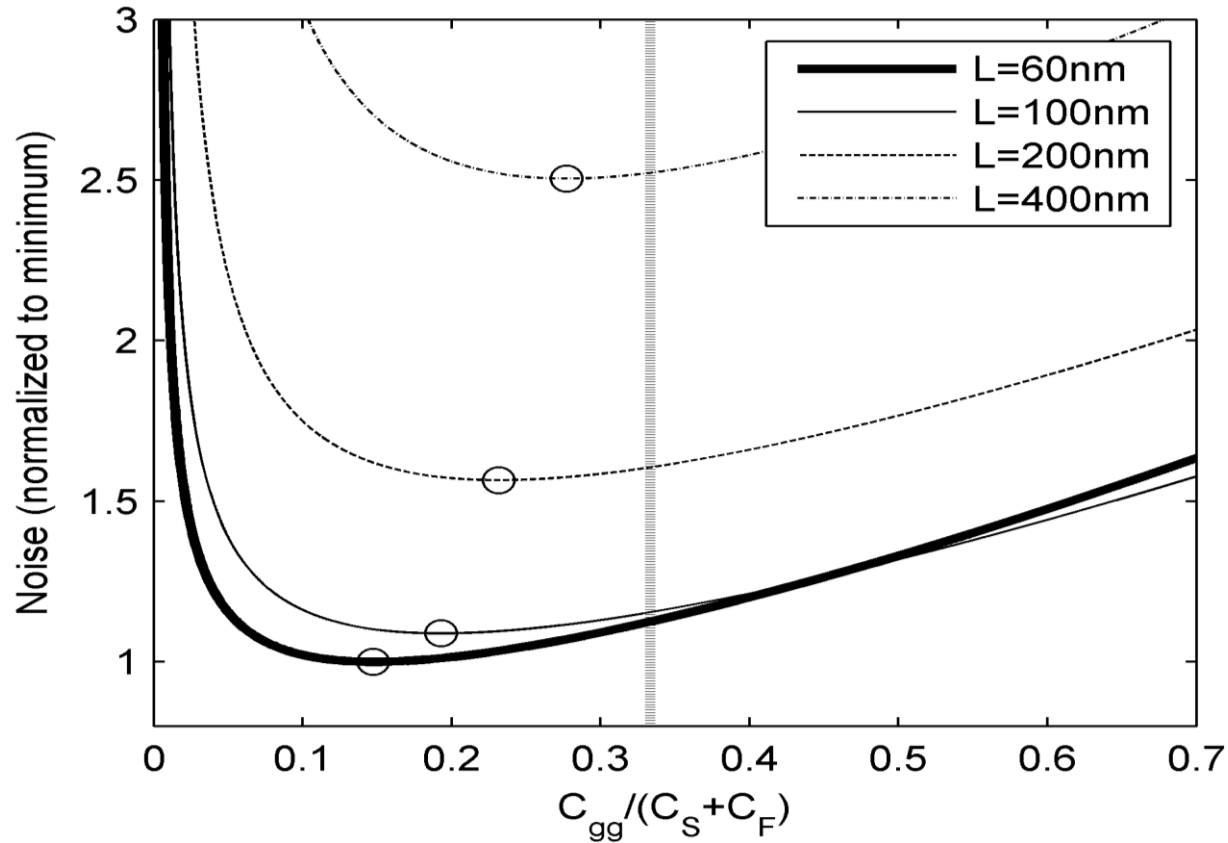
% Compute relative noise level

```
Cgg = [W; W; W; W].*lookup(nch, 'CGG_W', 'ID_W', ID./W, 'L', L);
```

```
gm = [W; W; W; W].*lookup(nch, 'GM_W', 'ID_W', ID./W, 'L', L);
```

```
Noise = (Cf_plus_Cs + Cgg).^2./gm;
```

Result



- Optimum approaches square-law solution for long channels, but is far off for short channel

Scenario 3: Optimization Assuming Constant Noise and BW

- This is the scenario most frequently encountered in practice
- Analysis shows that

$$I_D \propto \left(\frac{1}{1 - \frac{C_B}{C_F} \frac{\omega_c}{\omega_T}} \right)^2 \cdot \frac{I_D}{g_m} \quad \begin{array}{l} \omega_c \rightarrow \text{BW spec} \\ C_B \rightarrow \text{Buffer input capacitance} \end{array}$$

- Minimizing the current boils down to optimizing the tradeoff between g_m/I_D and ω_T
- First-order optimum assuming square law ($g_m/I_D \cdot \omega_T = \text{const.}$)

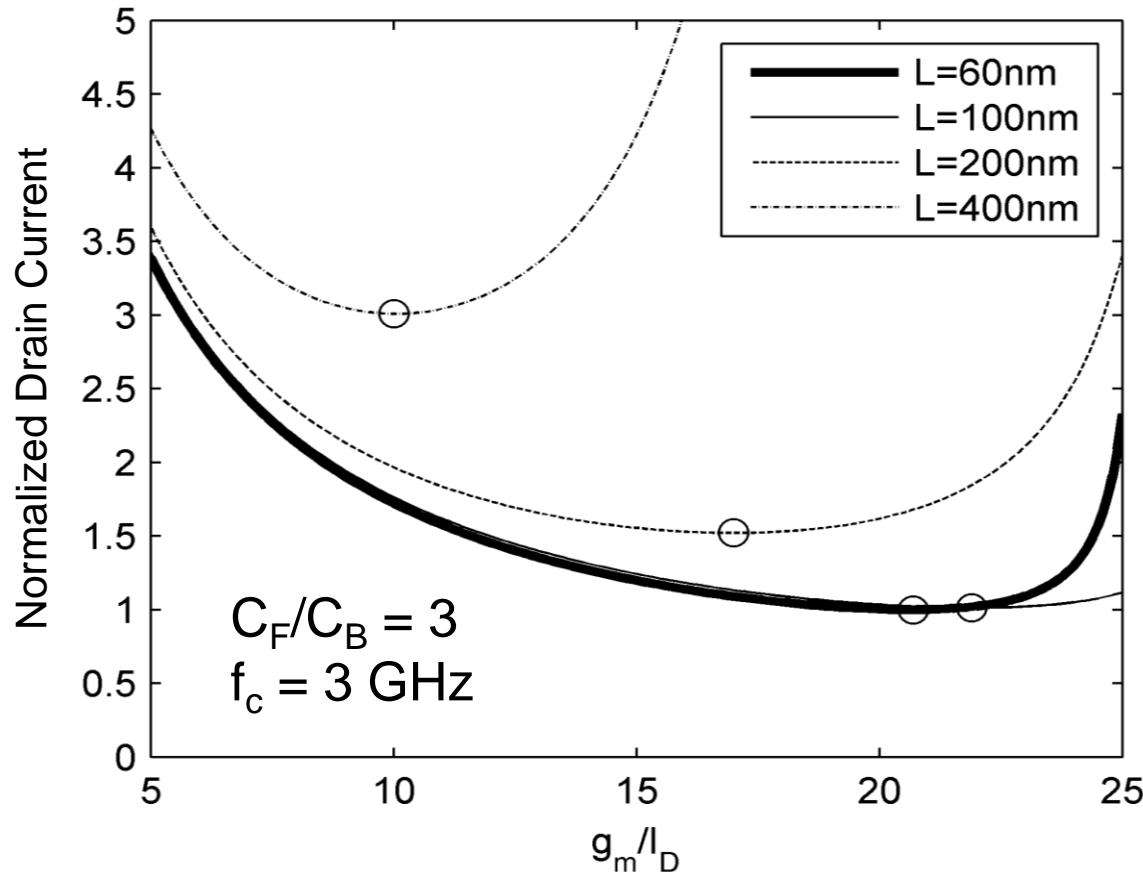
$$\omega_{T,opt} = 3 \frac{C_B}{C_F} \omega_c$$

$$C_{gg} = \frac{C_S + C_F}{2}$$

Example: Actual Optima Using Lookup Table Data

$gm_ID = 5:0.1:25;$

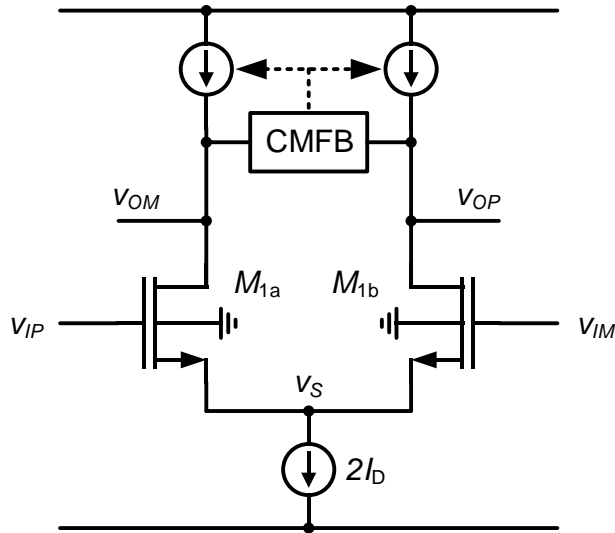
$wT = \text{lookup}(\text{nch}, 'GM_CGG', 'GM_ID', gm_ID, 'L', L);$



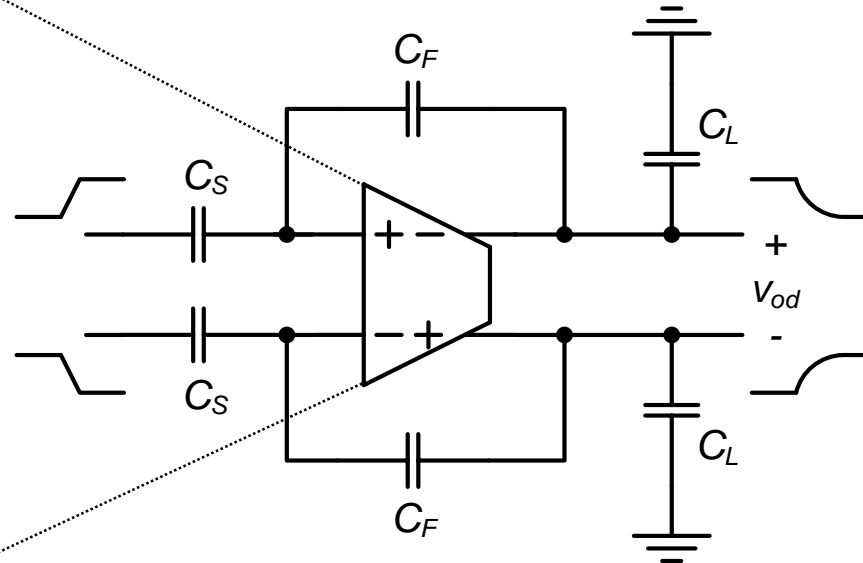
Optimum Parameters			
L (nm)	g_m/I_D (S/A)	f_T (GHz)	$C_{gg}/(C_S+C_F)$
60	20.7	9.6	0.116
100	21.9	7.5	0.155
200	17.0	5.1	0.247
400	10.0	3.9	0.345

Actual optima are far from square law prediction

Basic OTA for Switched Capacitor Circuit



(Switches not shown)



$$\overline{v_{od}^2} = 2 \frac{\gamma_n}{\beta} \frac{k_B T}{C_{Ltot}}$$

$$C_{Ltot} = C_L + (1 - \beta)C_F$$

$$\beta = \frac{C_F}{C_F + C_S + C_{gs}}$$

$$\beta_{max} = \frac{C_F}{C_F + C_S}$$

Optimization

- Design equations are similar to charge amplifier
 - Main difference is self-loading by feedback network
- Analysis shows

$$I_D \propto \left(\frac{1 + \frac{C_S}{C_F} - \frac{\omega_c}{\omega_{Ts}}}{1 - \left(1 + \frac{C_L}{C_F}\right) \frac{\omega_c}{\omega_{Ts}}} \right)^2 \frac{I_D}{g_m} \quad \omega_{Ts} = \frac{g_m}{C_{gs}}$$

- Optimum assuming square law

$$\omega_{Ts,opt} = 3 \frac{C_L}{C_F} \omega_c + 3$$

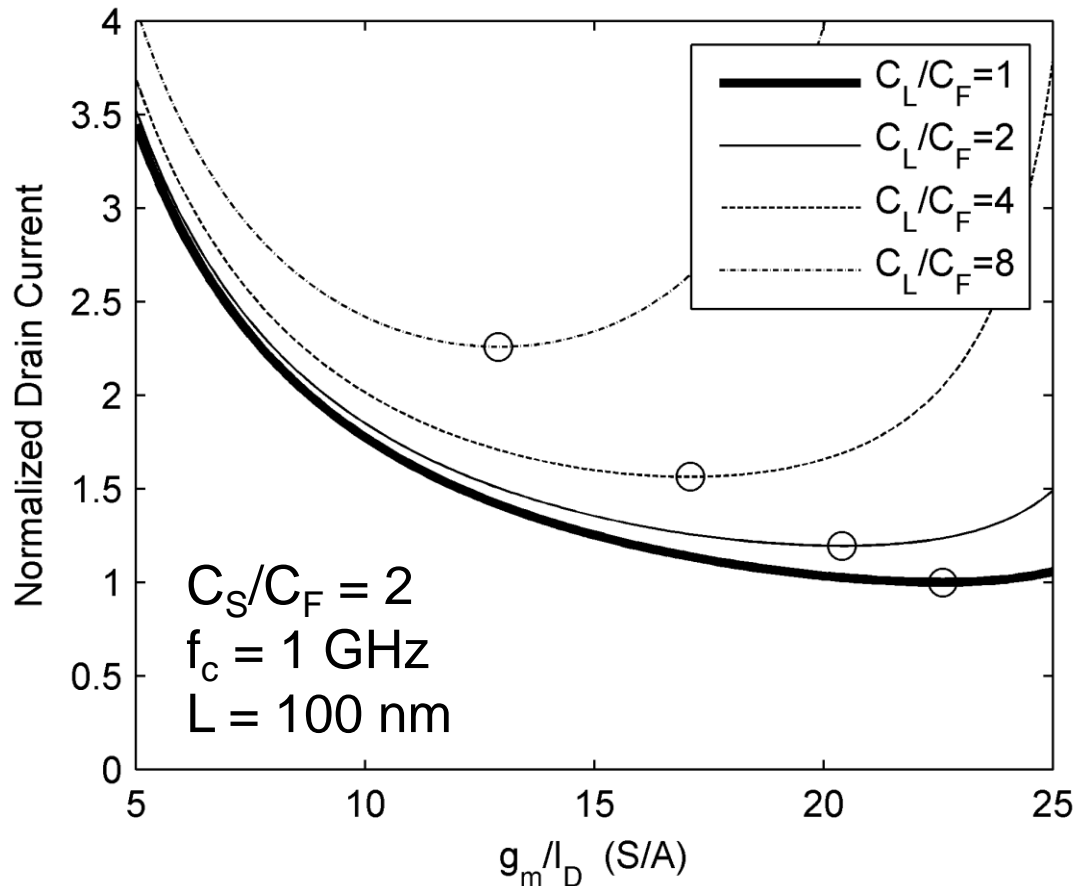
$$C_{gs,opt} = \frac{C_S + C_F}{2}$$

$$\frac{\beta_{opt}}{\beta_{max}} = \frac{3}{4}$$

Example: Actual Optima Using Lookup Table Data

`gm_ID = 5:0.1:25;`

`wTs = lookup(nch, 'GM_CGS', 'GM_ID', gm_ID, 'L', L);`

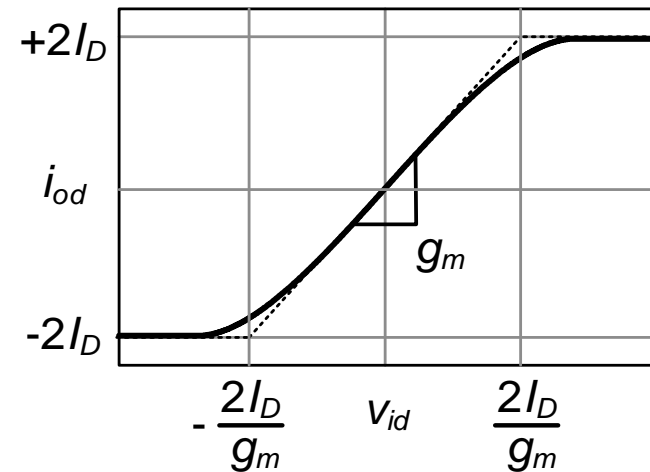
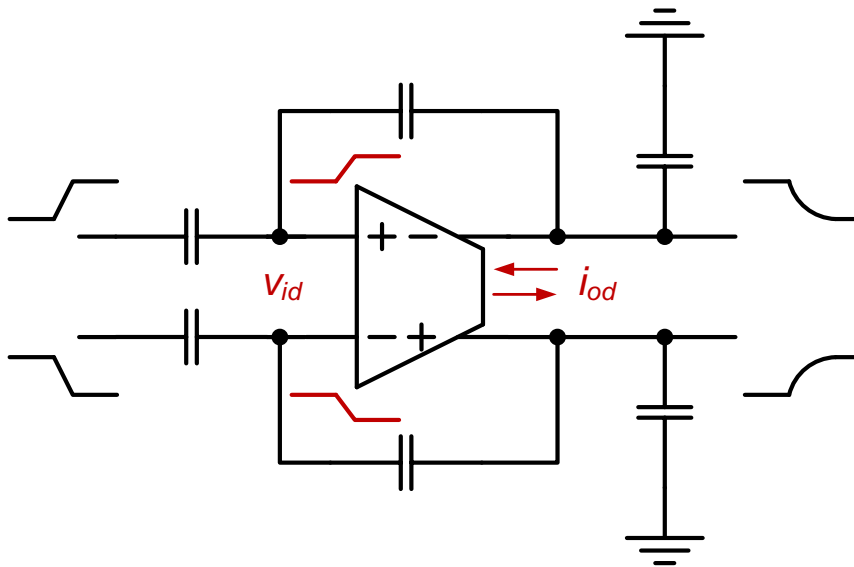


C_L/C_F	Optimum Parameters		
	g_m/I_D (S/A)	f_{Ts} (GHz)	β/β_{max}
1	22.6	12.0	0.857
2	20.4	15.6	0.825
4	17.1	22.3	0.788
8	12.9	35.6	0.755

Optimum shifts to higher inversion levels for larger C_L

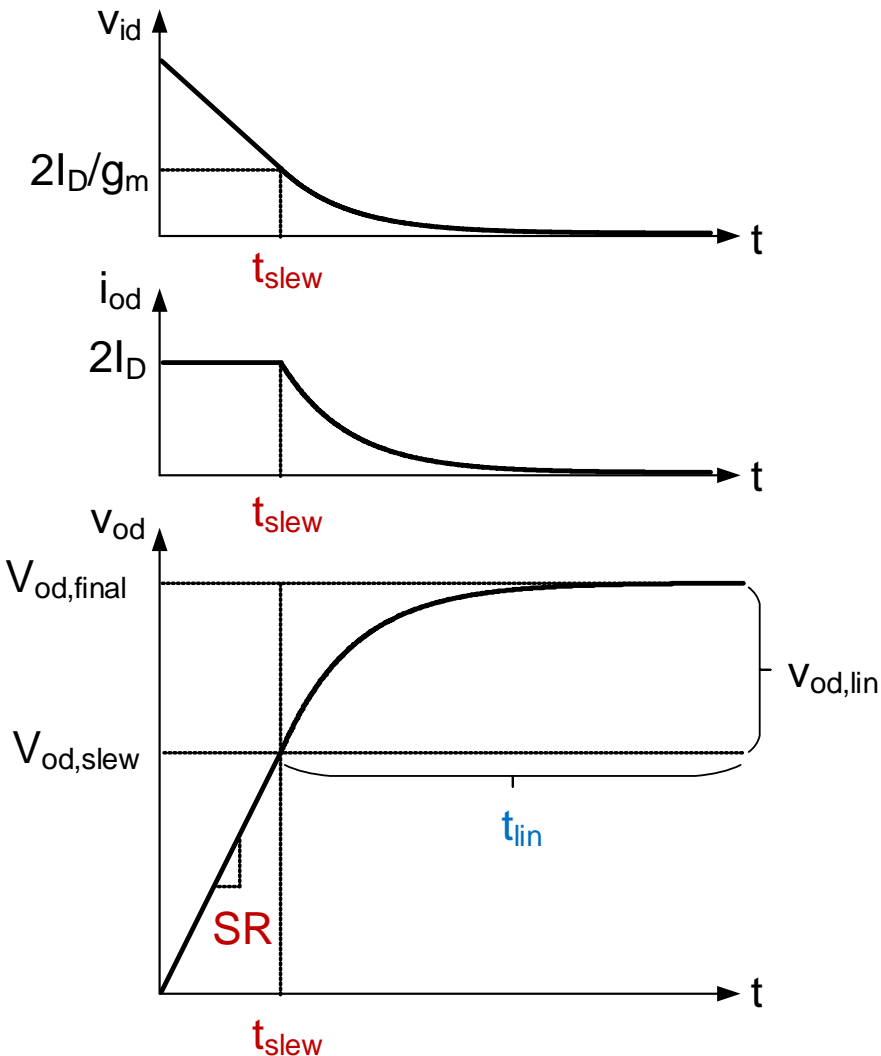
Slewing

- The preceding results all assumed small-signal operation
- In reality, switched capacitor circuits suffer from slewing



- The input pair leaves its linear region during the initial transient

Settling Time



Linear Settling Time Constant

$$\tau = \frac{C_{Ltot}}{\beta g_m}$$

Slew Rate

$$SR = \frac{2I_D}{C_{Ltot}} = \frac{2I_D}{\tau \beta g_m}$$

Total Settling Time (ϵ_d = dynamic error)

$$t_s = t_{slew} + t_{lin} = \tau(X - 1 - \ln(\epsilon_d X))$$

$$X = \frac{v_{od,final}}{v_{od,lin}} = v_{od,final} \cdot \frac{\beta g_m}{2 I_D}$$

Optimization

- There is no longer a closed form equation that can relate the design specs to the drain current
 - But, we can still solve this problem using lookup tables!
1. Sweep g_m/I_D
 2. For each g_m/I_D and for a range of β , compute the following:
 - a) C_{Ltot} based on noise specification
 - b) X , I_D , g_m
 - c) C_{gs} based on f_T and g_m
 - d) Actual β value
 3. Now find the self-consistent values of β
 4. Plot I_D versus g_m/I_D at the self-consistent points

Sample Matlab Code

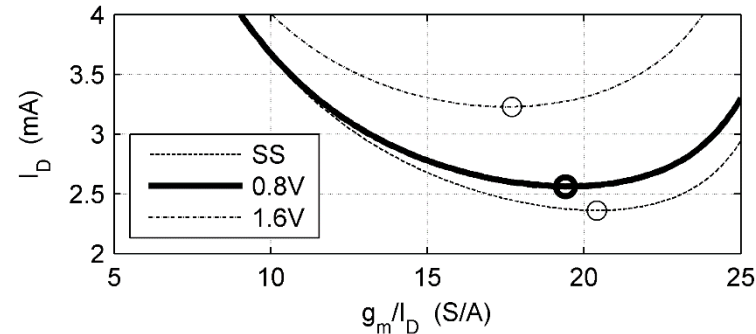
```
% Sweep range for gm/ID and beta
gm_ID = 5:0.1:25;
beta_sweep = linspace(0.25*beta_max, beta_max, 1000);
wts = lookup(nch, 'GM_CGS', 'GM_ID', gm_ID, 'L', L);

for i = 1:length(vodfinal);
    for j = 1:length(gm_ID)
        % compute CLtot based on noise specification
        CLtot = 2*kB*T*gamma./beta_sweep/vod_noise^2;

        % compute ratio X = vodfinal/vodlin, current and component sizes
        X = vodfinal(i)*beta_sweep/2*gm_id(j); X(X<1) = 1;
        ID = CLtot./beta_sweep/gm_id(j)/ts.*(X-1 - log(ed*X));
        gm = gm_id(j)*ID; Cgs = gm/wts(j); CF = CLtot./(CL_CF + 1-beta_sweep);

        % compute actual beta and find self-consistent point
        beta_actual = CF./(CF*(1+CS_CF) + Cgs);
        d = beta_actual - beta_sweep;
        idx = find(d(1:end-1).*d(2:end)<0);
        if (idx) ID_valid(j,i) = ID(idx); end
    end
end
```

Example ($v_{od,final} = 10\text{mV (SS)}, 0.8\text{V}, 1.6\text{V}$)

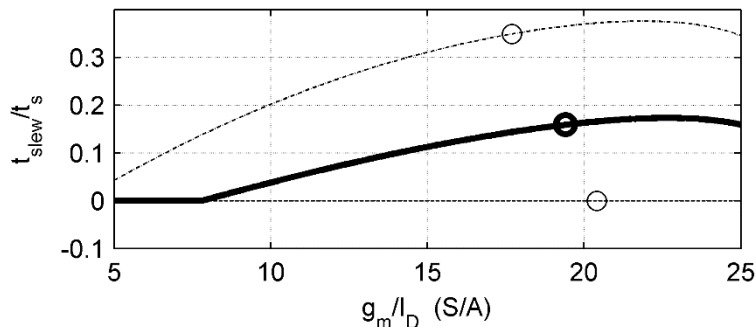
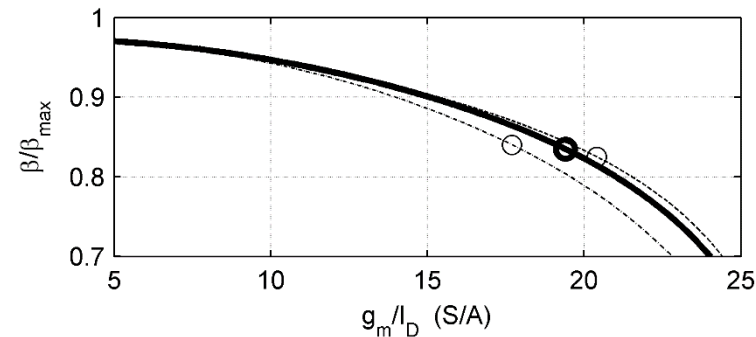


Parameters:

$$C_S/C_F = C_L/C_F = 2$$

$$t_s = 1.1 \text{ nS}$$

$$L = 100 \text{ nm}$$



- Optimum shifts to smaller g_m/I_D for larger signal ($v_{od,final}$)
- Optimum feedback factor stays close that of small-signal (SS) optimum
- Slewing time about 15...33%
 - Note: The 1.6V example is not practical; serves only illustrative purposes

Sizing

% Capacitance values

$$CL_{tot} = 2 \cdot k_B \cdot T \cdot \gamma_{\text{opt}} / \beta_{\text{opt}} / v_{od_noise}^2$$

$$C_F = CL_{tot} / (CL_{CF} + 1 - \beta_{\text{opt}})$$

$$C_S = C_{S_CF} \cdot C_F$$

$$C_L = CL_{CF} \cdot C_F$$

% Device size

$$JD = \text{lookup}(\text{nch}, 'ID_W', 'GM_ID', g_{m_ID_opt}, 'L', L)$$

$$W = ID_{\text{opt}} / JD$$

Results (for $v_{od,final} = 0.8V$):

$$C_{L_{tot}} = 2.06 \text{ pF}$$

$$C_L = 1.52 \text{ pF}$$

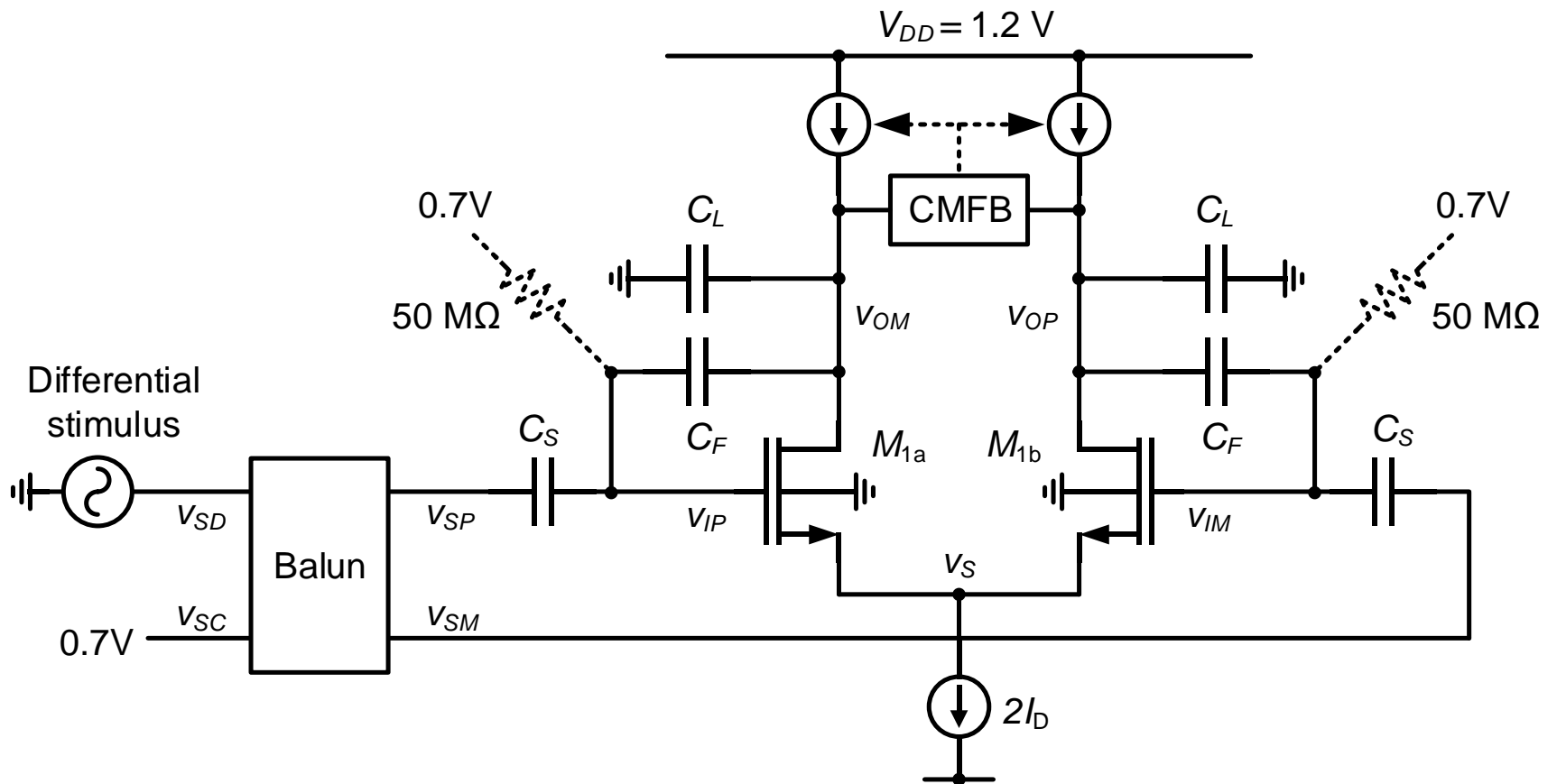
$$C_F = 0.76 \text{ pF}$$

$$C_S = 1.52 \text{ pF}$$

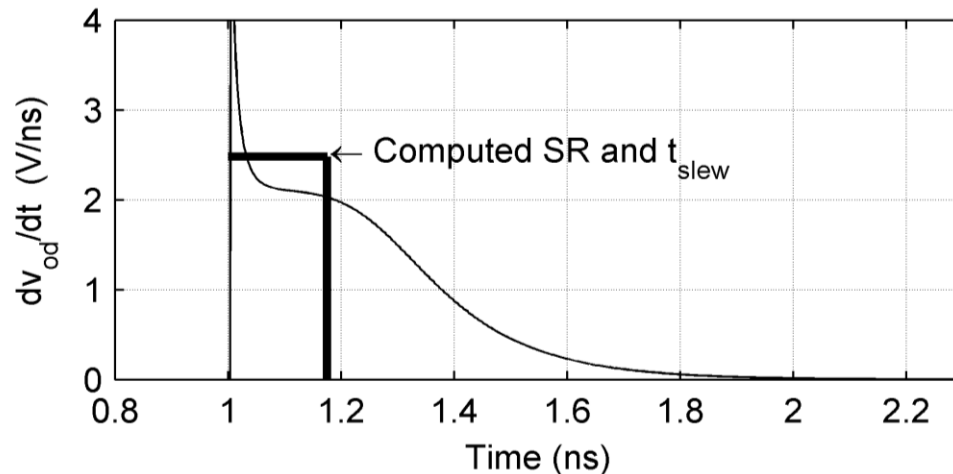
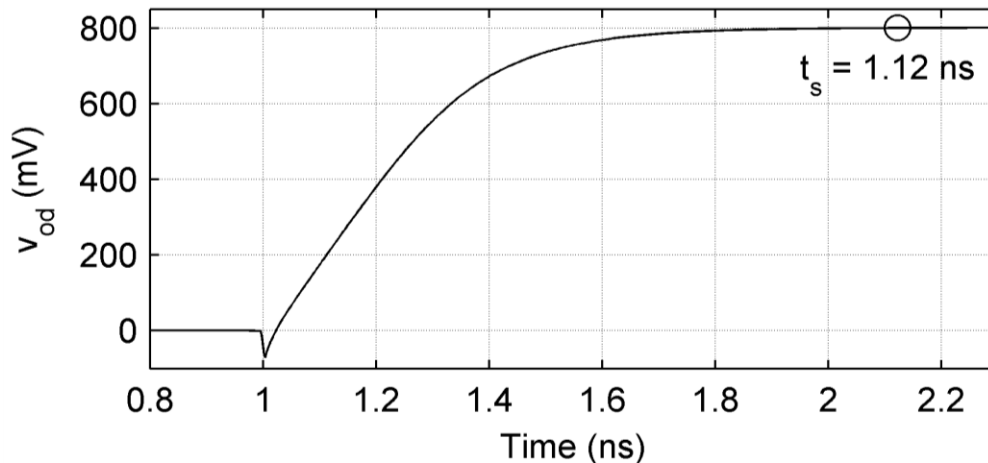
$$I_D = 2.6 \text{ mA}$$

$$W = 692 \text{ } \mu\text{m}$$

Spice Testbench

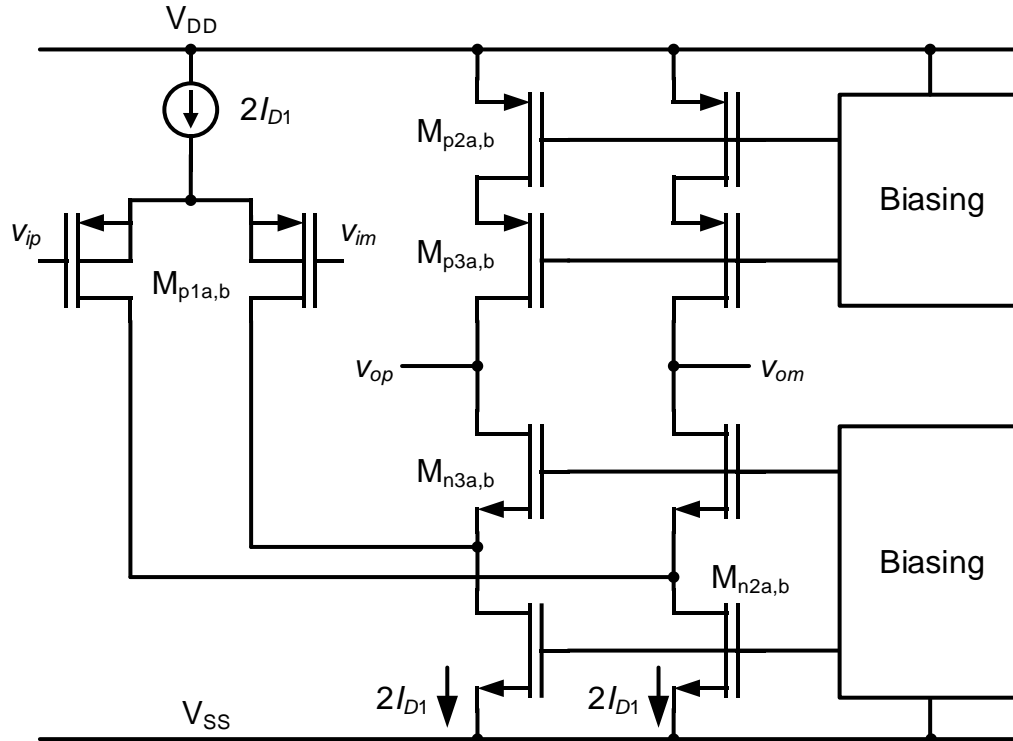


Simulation Result

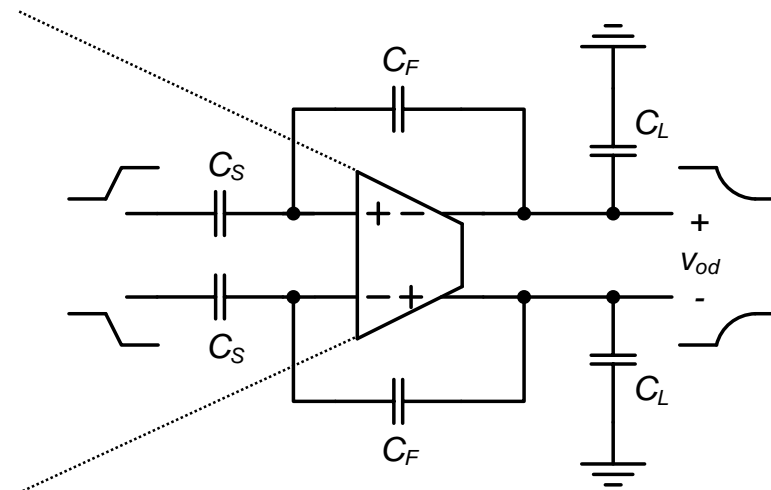


- Slew rate somewhat smaller than expected
 - Junction caps (neglected in design script)
 - Differential pair does not steer fully
- Linear portion somewhat faster than expected due to r_{ds} (neglected in design script)
- Final result is very close to target

(CMFB not shown)



(Switches not shown)



- + Higher DC gain than basic OTA
- Nondominant pole (f_{p2}), reduced output swing

Specifications

Parameter	Value	Comment
C_s/C_F	2	Closed-loop gain
C_L/C_F	1	Near optimum for pipeline ADC
Output swing	800 mV _{pp}	Differential
Total integrated output noise	400 μ V _{rms} (DR = 57 dB)	Differential
Phase margin	75°	For optimum transient settling
Dynamic settling error	0.1 %	
DC loop gain	> 50	Can use gain boosting if more gain is needed
Settling time	As small as possible (limited by f_{p2})	Ignore slewing (for simplicity)
Power dissipation	Minimize	

Design Equations



Feedback Factor

$$\beta = \frac{C_F}{C_F + C_S + C_{in}}$$

$$C_{in} \cong C_{ggp1} + C_{gdp1} \frac{g_{mp1}}{g_{mn3}}$$

Time Constant

$$\tau \cong \frac{C_{Ltot}}{\beta \kappa g_m}$$

$$C_{Ltot} = C_L + (1 - \beta)C_F + C_{self}$$

Self loading
(junction cap)

$$\kappa \cong \frac{g_{mn3}}{g_{mn3} + g_{dsp1} + g_{dsn2}}$$

Current divider at cascode

DC Loop Gain

$$RR_0 = \beta \kappa g_{mp1} R_o$$

$$\frac{1}{RR_0} \cong \frac{1}{\beta \kappa} \left(\frac{1}{\left(1 + \frac{g_{m2,3}}{g_{dsp2}}\right) \frac{g_{m2,3}}{g_{dsp3}}} + \frac{1}{\left(1 + \frac{g_{m2,3}}{3g_{dsn2}}\right) \frac{g_{m2,3}}{g_{dsn2}}} \right)$$

Noise

$$\overline{v_{od}^2} = \frac{\alpha k_B T}{\beta C_{Ltot}}$$

$$\alpha = 2\gamma_n \left(1 + \frac{(g_m/I_D)_{p2}}{(g_m/I_D)_{p1}} + 2 \frac{\gamma_p (g_m/I_D)_{n2}}{\gamma_n (g_m/I_D)_{p1}} \right)$$

Design Equations

Nondominant Pole

$$\omega_{p2} \cong \frac{g_{mn3} + g_{mbn3}}{C_{ddp1} + C_{ssn3} + C_{ddn2}} \cong \frac{g_{mn3} + g_{mbn3}}{C_{ssn3} + 3C_{ddn3}}$$

- Equation set is complex and tangled up
- No hope to find a closed-form solution
- Approach
 - First identify hard constraints to simplify the problem
 - Next “untangle” the equations using numerical sweep
 - Very similar to previous example on slewing

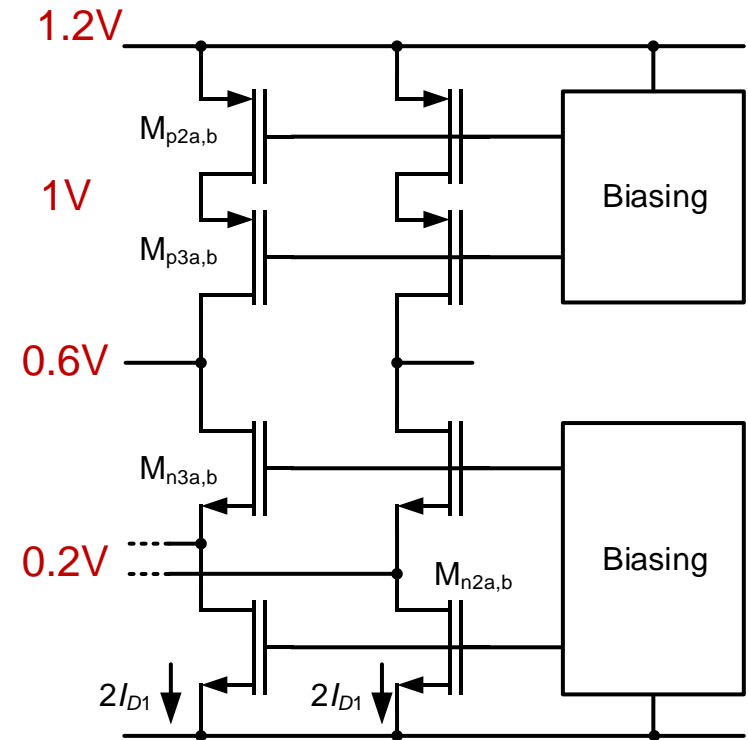
Swing Constraint

Swing $\sim \pm 400\text{mV}$
Differential

Decision:

$$\left(\frac{g_m}{I_D}\right)_{n2} = \left(\frac{g_m}{I_D}\right)_{n3} = \left(\frac{g_m}{I_D}\right)_{p2} = \left(\frac{g_m}{I_D}\right)_{p3} = 15 \text{ S/A}$$

$$\Rightarrow V_{Dsat} \cong \frac{2}{\frac{g_m}{I_D}} = 133\text{mV}$$



Gain Constraint

$$\frac{1}{RR_0} \cong \frac{1}{\beta \kappa} \left(\frac{1}{\left(1 + \frac{g_{m2,3}}{g_{dsp2}}\right) \frac{g_{m2,3}}{g_{dsp3}}} + \frac{1}{\left(1 + \frac{g_{m2,3}}{3g_{dsn2}}\right) \frac{g_{m2,3}}{g_{dsn2}}} \right)$$

gm_ID23 = 15;

% Conservative estimate for kappa

kappa_cons = 0.7;

% Reasonable estimate for beta

beta_max = 1/(1+CS_CF);

beta_guess = 2/3*beta_max;

% Channel length sweep

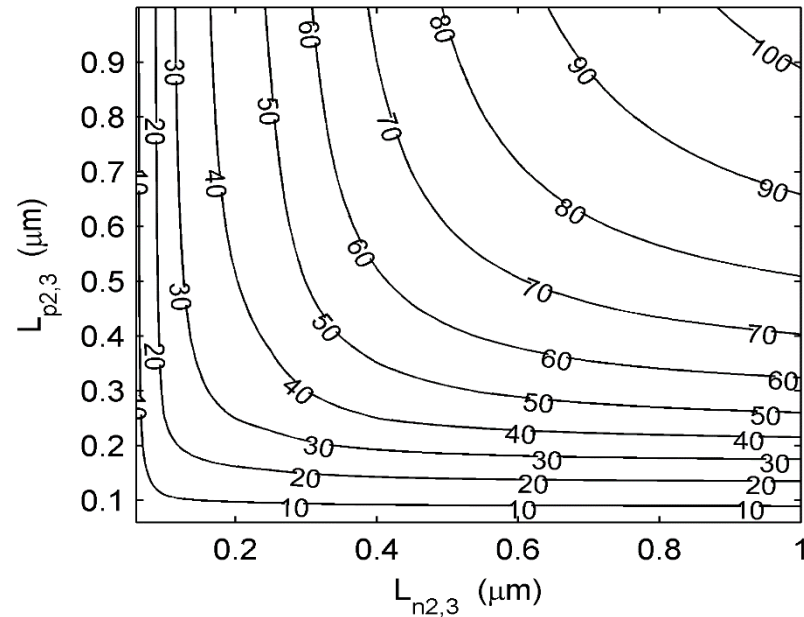
L = linspace(0.06, 1);

gm_gds_p2 = lookup(pch, 'GM_GDS', 'GM_ID', gm_ID23, 'VDS', 0.2, 'L', L);

gm_gds_p3 = lookup(pch, 'GM_GDS', 'GM_ID', gm_ID23, 'VDS', 0.4, 'L', L);

gm_gds_n3 = lookup(nch, 'GM_GDS', 'GM_ID', gm_ID23, 'VDS', 0.4, 'L', L);

gm_gds_n2 = lookup(nch, 'GM_GDS', 'GM_ID', gm_ID23, 'VDS', 0.2, 'L', L);



Decision:

$L_{n2,3} = 400 \text{ nm}$

$L_{p2,3} = 400 \text{ nm}$

Nondominant Pole & Settling Time Estimates

% Chosen lengths

Ln23 = 0.4; Lp23 = 0.4;

% Resulting device parameters

gmb_gm_n3 = lookup(nch, 'GMB_GM', 'GM_ID', 15, 'VDS', 0.4, 'L', Ln23);

gm_css_n3 = lookup(nch, 'GM_CSS', 'GM_ID', 15, 'VDS', 0.4, 'L', Ln23);

cdd_css_n3 = lookup(nch, 'CDD_CSS', 'GM_ID', 15, 'VDS', 0.4, 'L', Ln23);

% Nondominant pole

wp2 = 1/2/pi * gm_css_n3 * (1+gmb_gm_n3)/(1 + 3*cdd_css_n3)

% Set gain-bandwidth product for proper phase margin

fc = fp2/4

% Settling time estimate

tau = 1/2/pi/fc

epsilon_d = 1e-3;

ts = -tau*log(epsilon_d)

Results:

$f_{p2} = 1.74 \text{ GHz}$

$f_c = 435 \text{ MHz}$

$t_s = 2.5 \text{ ns}$

Sweep for Current Optimization

1. Sweep g_m/I_D of the input pair
2. For each g_m/I_D and for a range of β , compute
 - a) Excess noise factor α
 - b) C_{Ltot} based on noise specification*
 - c) Factor κ
 - d) Input pair transconductance
 - e) C_{in} , C_f , and actual value of β
3. Find the self-consistent values of β
4. Plot I_D versus g_m/I_D for the self consistent points

***Caveat:** We do not know C_{self} (part of C_{Ltot}) at this point; start by assuming $C_{self} = 0$ and refine later

Matlab Script

```
% Sweep parameters
Cself_est = 0;
Lp1 = [0.06 0.1 0.2 0.3]; gm_ID1 = 5:0.1:22;
beta_sweep = linspace(0.25*beta_max, beta_max, 200);

for i = 1:length(Lp1)
    % Compute input pair device parameters for each length
    wt1 = lookup(pch, 'GM_CGG', 'GM_ID', gm_ID1, 'L', Lp1(i));
    gm_gds1 = lookup(pch, 'GM_GDS', 'GM_ID', gm_ID1, 'L', Lp1(i));
    Cgd_Cgg1 = lookup(pch, 'CGD_CGG', 'GM_ID', gm_ID1, 'L', Lp1(i));

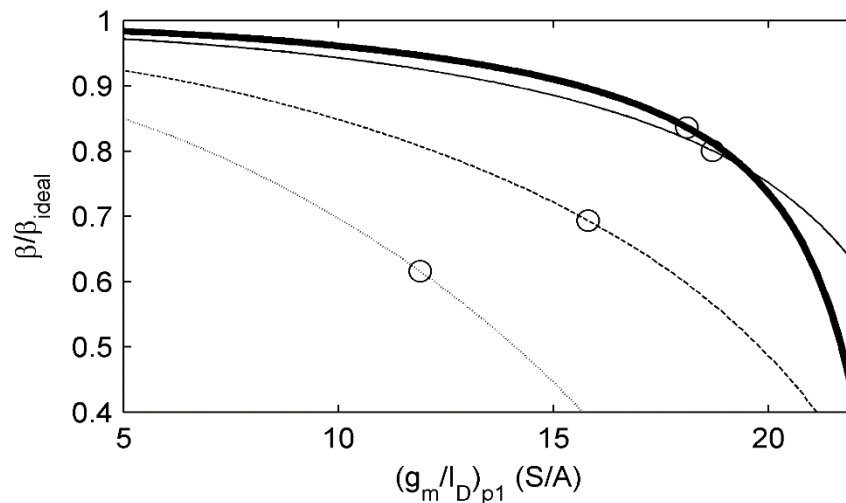
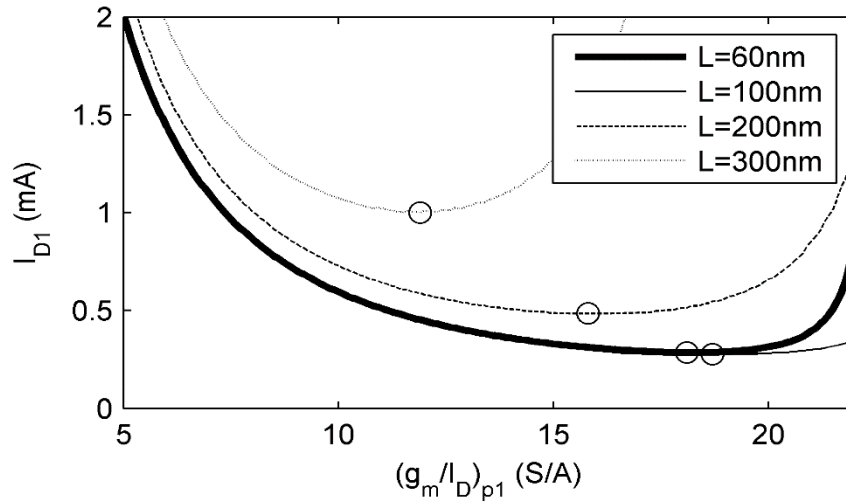
    for j = 1:length(gm_ID1)
        % Compute excess noise factor and total load to meet the noise spec
        alpha = 2*gamma_n*(1+gm_id23/gm_ID1(j)+2*gamma_p/gamma_n*gm_id23/gm_ID1(j));
        CLtot = alpha./beta_sweep*kBoltzmann*nch.TEMP/vod_noise^2;

        % Compute required gm to meet the gain-bandwidth requirement
        kappa = 1/(1+ 1/gm_gds1(j)*gm_ID1(j)/gm_id23 + 2/gm_gds_n3);
        gm1 = 2*pi*fc*CLtot./beta_sweep/kappa;

        % Compute the amplifier's input capacitance and resulting beta
        Cin = gm1/wt1(j)*(1 + Cgd_Cgg1(j)*gm_ID1(j)/gm_id23); CF = (CLtot - Cself_est)./(CL_CF+1-beta_sweep);
        beta_actual = CF./(CF+CS_CF*CF+Cin);

        % Find self-consistent beta values and store parameters
        d = beta_actual - beta_sweep; idx = find(d(1:end-1).*d(2:end)<0);
        if(idx) ID1_valid(i,j) = gm1(idx) / gm_ID1(j); end
    end
end
end
```


Result



- Decide to use $L = 100\text{ nm}$
- $I_D = 276\text{ }\mu\text{A}$, $g_m/I_D = 18.7\text{ S/A}$
- Can now size all devices and estimate C_{self}

Sizing and C_{self} Estimate

% Sizing

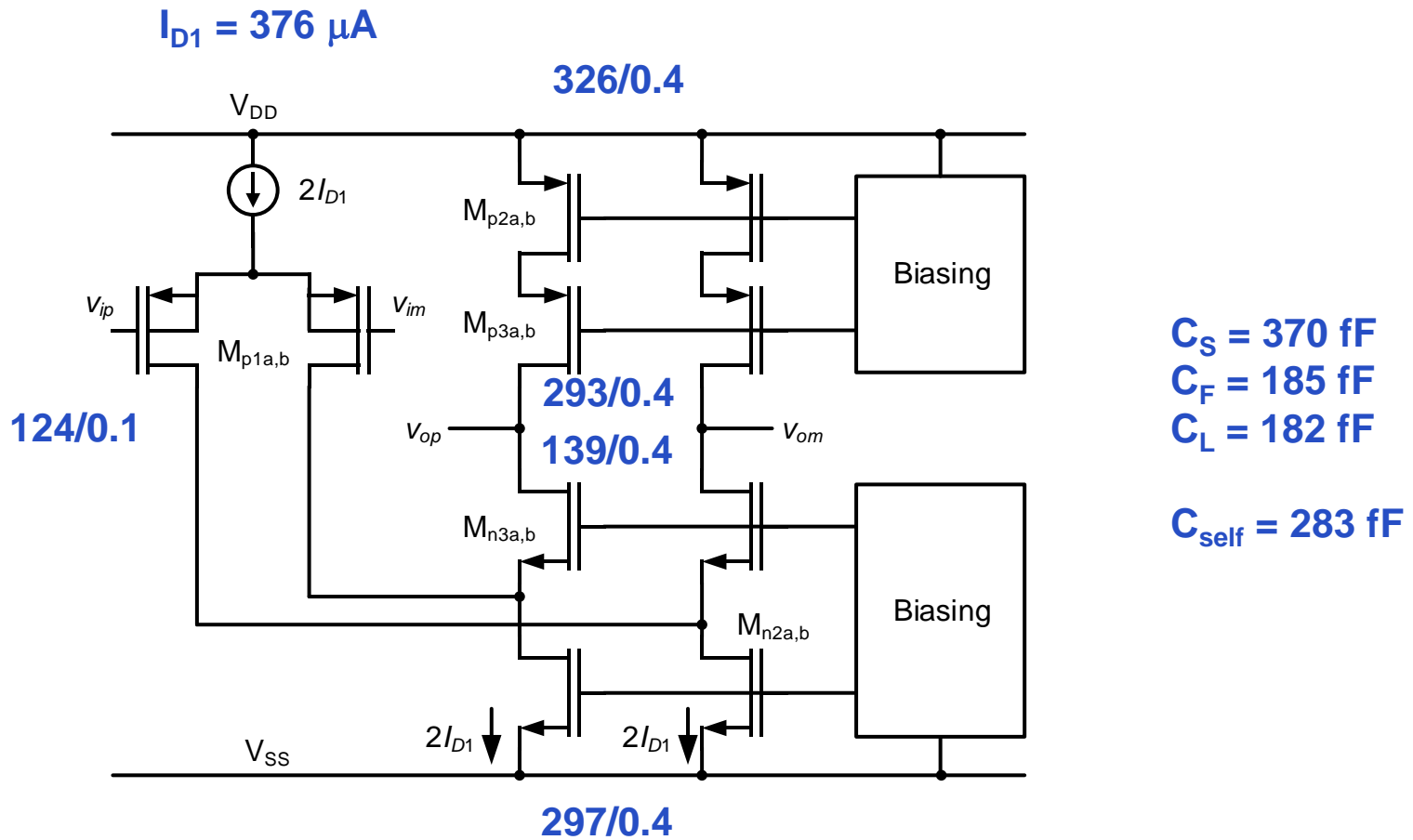
```
Wp1 = ID1_chosen/lookup(pch, 'ID_W', 'GM_ID', gm_id1_chosen, 'L', Lp1)
Wp2 = ID1_chosen/lookup(pch, 'ID_W', 'GM_ID', gm_id23, 'VDS', 0.2, 'L', Lp23)
Wp3 = ID1_chosen/lookup(pch, 'ID_W', 'GM_ID', gm_id23, 'VDS', 0.4, 'L', Lp23)
Wn2 = 2*id1_chosen/lookup(nch, 'ID_W', 'GM_ID', gm_id23, 'VDS', 0.2, 'L', Ln23)
Wn3 = ID1_chosen/lookup(nch, 'ID_W', 'GM_ID', gm_id23, 'VDS', 0.4, 'L', Ln23)
```

% Estimate Cself

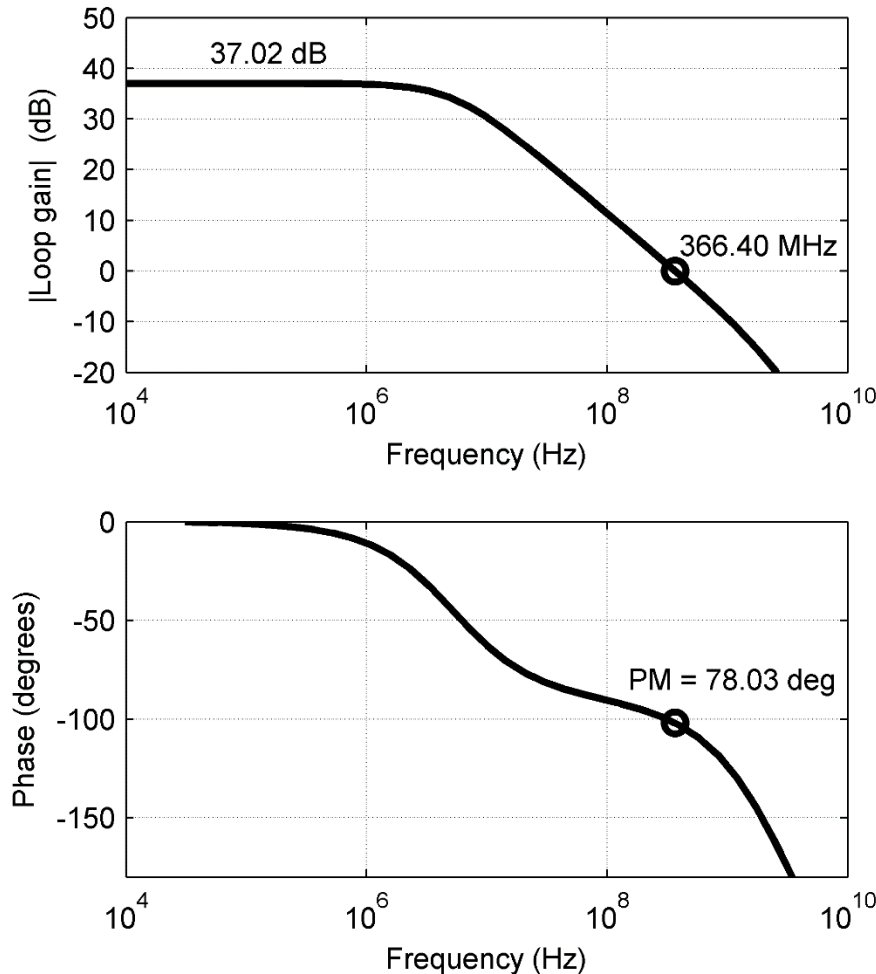
```
Cddp3 = Wp3*lookup(pch, 'CDD_W', 'GM_ID', gm_id23, 'VDS', 0.4, 'VSB', 0.2, 'L', Lp23)
Cddn3 = Wn3*lookup(nch, 'CDD_W', 'GM_ID', gm_id23, 'VDS', 0.4, 'VSB', 0.2, 'L', Ln23)
Cself_actual = Cddp3 + Cddn3
```

- C_{self} turns out to be 208 fF, about 45% of C_{Ltot} → Significant!
- Run the script again with $C_{\text{self,est}} = 208 \text{ fF}$
 - And repeat if desired, until “convergence”...

Final Design

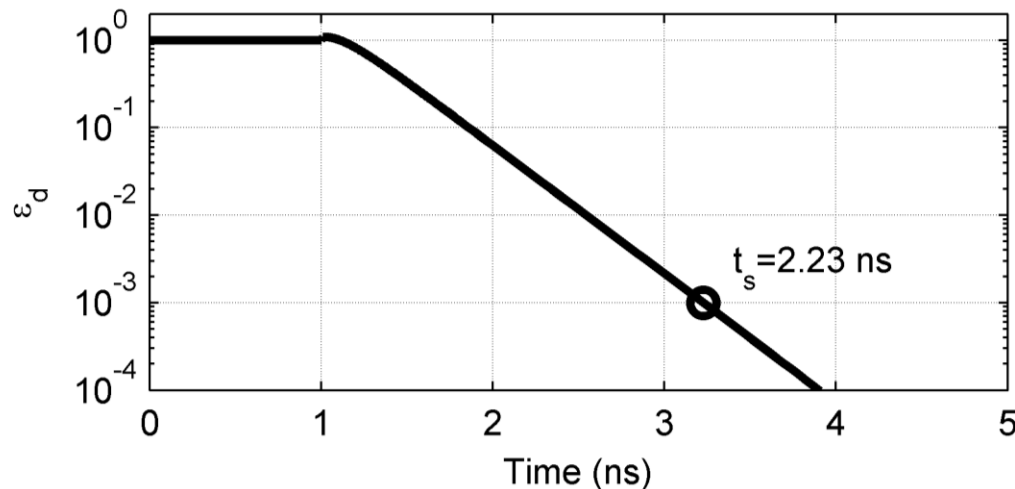
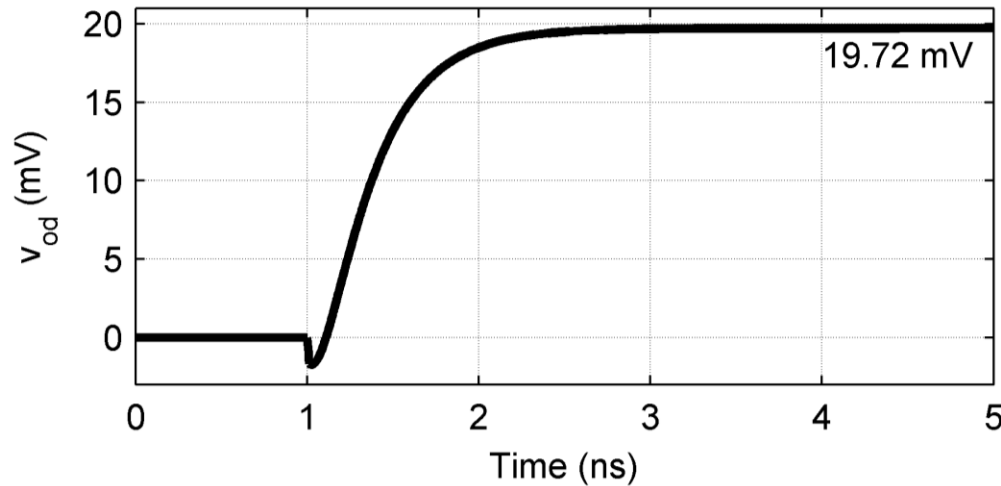


Loop Gain Simulation



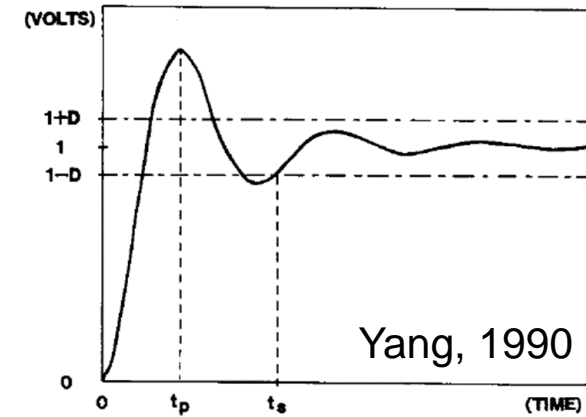
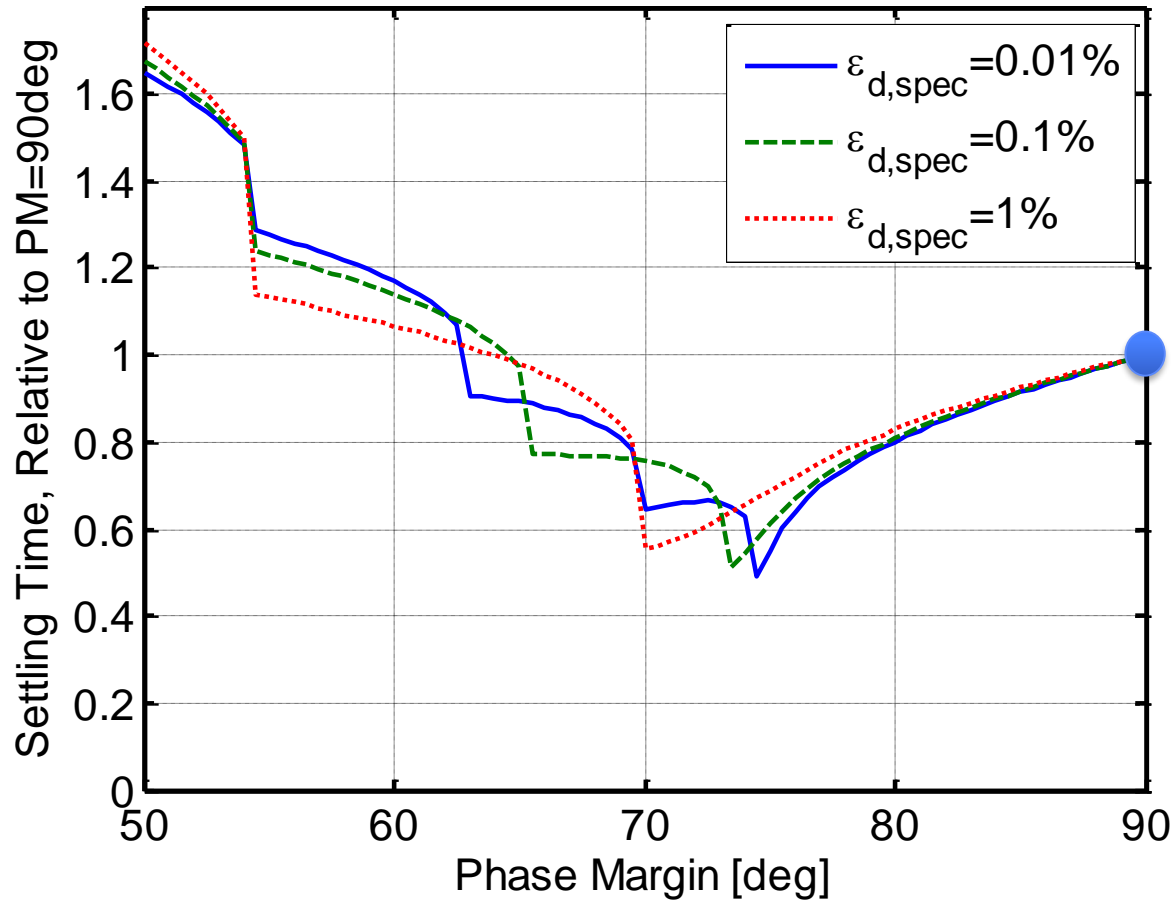
- Unity gain frequency smaller than targeted (435 MHz)
 - Remaining discrepancy in C_{self}
 - Impact of f_{p2}
- DC loop gain slightly larger than targeted (34 dB) due to conservative estimate
- Phase margin slightly larger than targeted due to reduced unity gain frequency

Transient Simulation



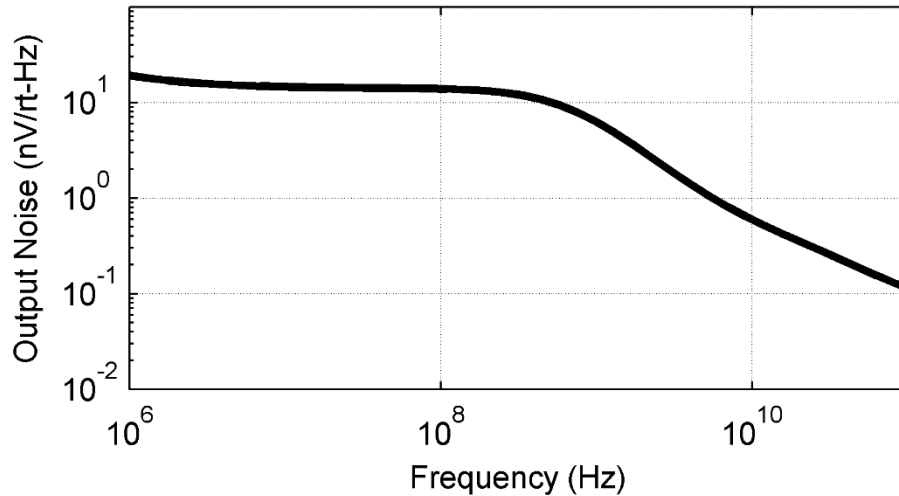
- Settling time is better than targeted (2.5 ns)
 - Due to non-dominant pole, which speeds up the transient
- Design meets settling requirements

Settling Time as a Function of Phase Margin

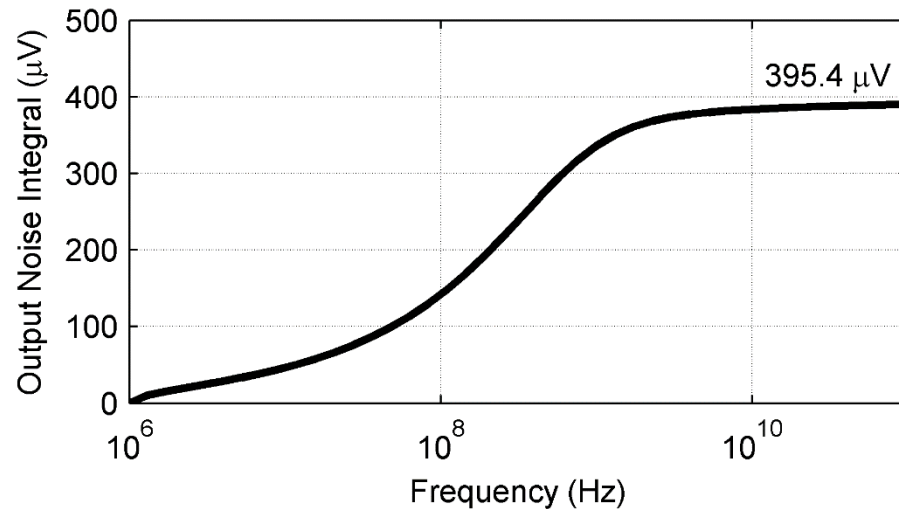


Single pole

Noise Simulation



- Almost exactly on target

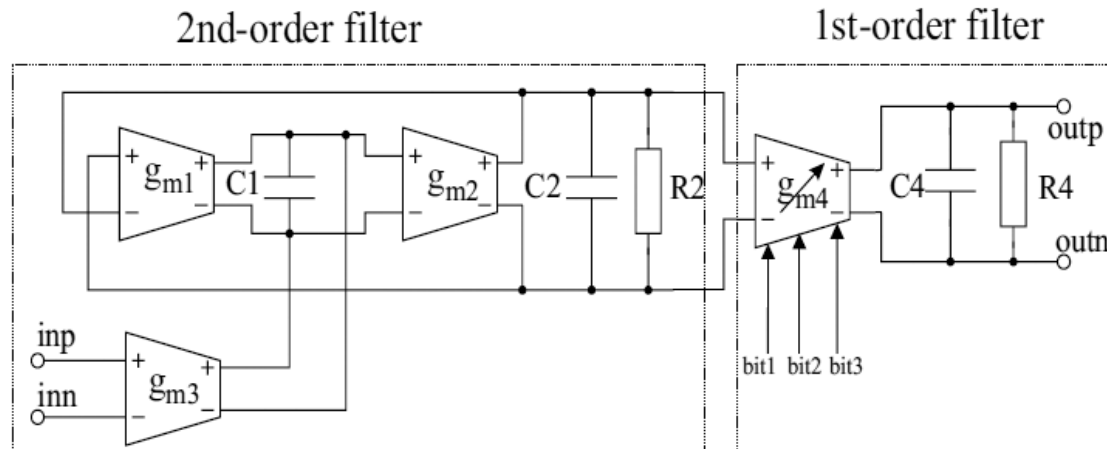


Possible Next Steps

- Errors are mostly due to first-order nature of design equations, which must omit second order effects to be useful
 - This is why we still run simulations...
- Design is very close to final specs and minor tweaking will get us very quickly to the final design point
- Next steps to consider
 - Try an n-channel input pair → simple change in code!
 - Revisit some of the initial design choices (e.g. channel lengths in output branch)
 - Include the effect of slewing if needed → small change in code!
 - Account for process variations → more later...

Distortion-Limited Gain Stages

- Several types of circuit are limited by nonlinearities
 - RF LNAs, g_m -C filters, etc.



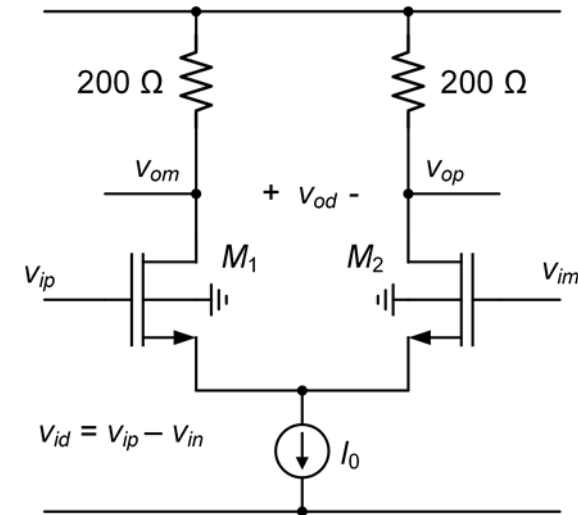
- It is possible to compute distortion based on g_m/I_D (for all inversion levels)
 - See Jespers & Murmann, ISCAS 2015

Example: Differential Amplifier

$$HD_3 = \frac{1}{24} \left(\frac{1}{nU_T} \right)^2 \frac{(1 + 3q)}{2(1 + 2q)^3} v_{id,pk}^2$$

$$q = \frac{1}{nU_T \cdot \frac{g_m}{I_D}} - 1$$

Valid for all inversion levels!



Parameter	Values			
$v_{id,pk}$ (mV), Spec	10	40	10	40
HD_3 (dB), Spec	-60.0	-60.0	-70.0	-70.0
g_m/I_D (S/A)	24.2	8.0	15.8	4.7
I_{TAIL} (mA), SPICE	0.826	2.51	1.26	4.25
W (μm), SPICE	321	25.7	82.7	18.1
HD_3 (dB), SPICE	-61.0	-60.4	-70.6	-67.5

Design with Process Corners

- Making circuits work across corners takes a significant amount of time (often dominates design time)
- No “magic bullet” that can cut this overhead to zero
 - But the g_m/I_D framework can help you think about corner behavior systematically

$\left\{ \frac{g_m}{I_D} \right\}$ transconductance efficiency

$\left\{ \frac{I_D}{W} \right\}$ drain current density

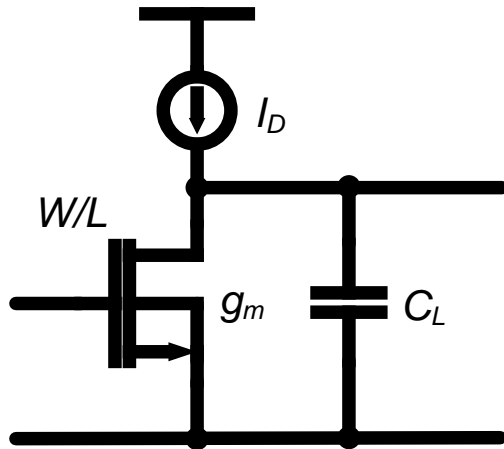
$\left\{ \frac{g_m}{g_{ds}} \right\}$ intrinsic gain

$\left\{ \frac{g_m}{C_{gg}} \right\}$ transit frequency

How do these parameters vary across corners?

Example: Intrinsic Gain Stage

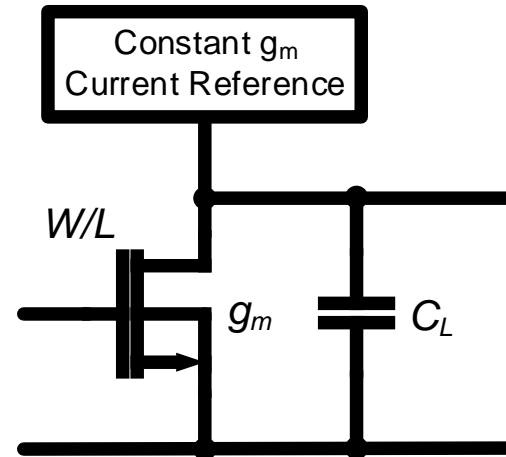
Biasing Scenario 1: Constant I_D



$$\frac{I_D}{W} = \text{const.} \quad GBW = \frac{g_m}{C_L} \neq \text{const.}$$

- + Moderate variations in V_{Dsat}
- Large variations in GBW

Biasing Scenario 2: Constant g_m

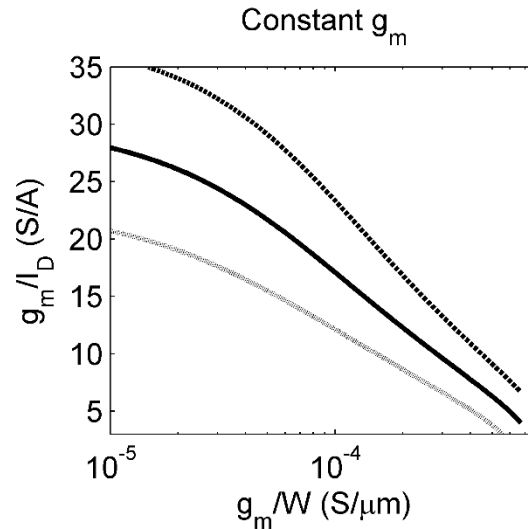
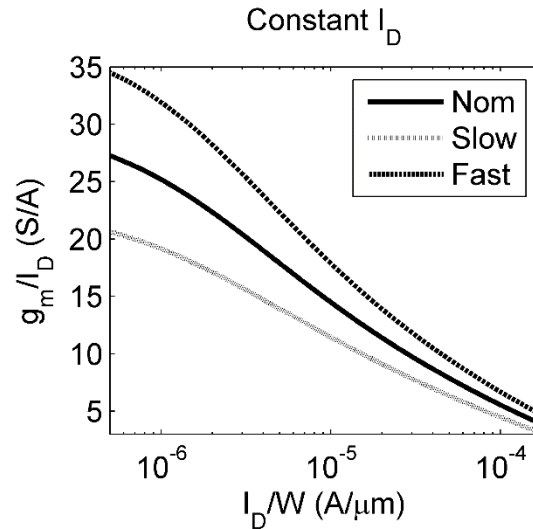


$$\frac{g_m}{W} = \text{const.} \quad GBW = \frac{g_m}{C_L} \approx \text{const.}$$

- + Small variations in GBW
- Large variations in I_D and V_{Dsat}

Practical designs often operate between these two extremes

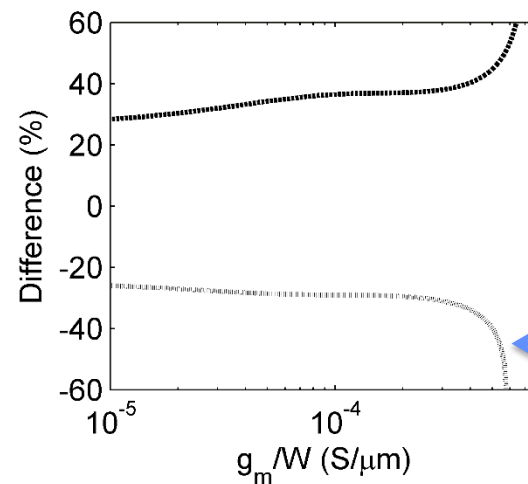
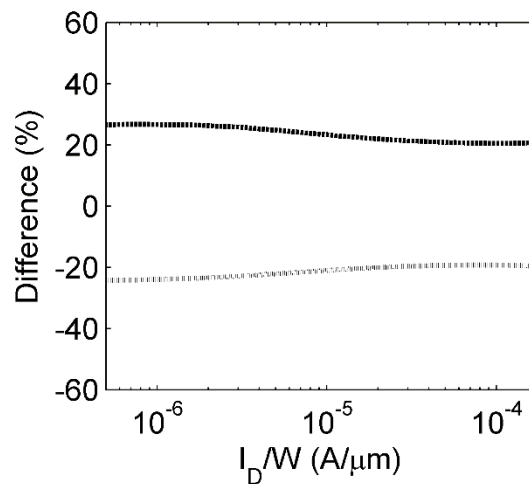
g_m/I_D Across Corners



n-Channel
 $L = 100\text{nm}$
 $V_{DS} = 0.6\text{ V}$

Temperatures:
 Fast = -40°
 Nom = 27°
 Slow = 125°

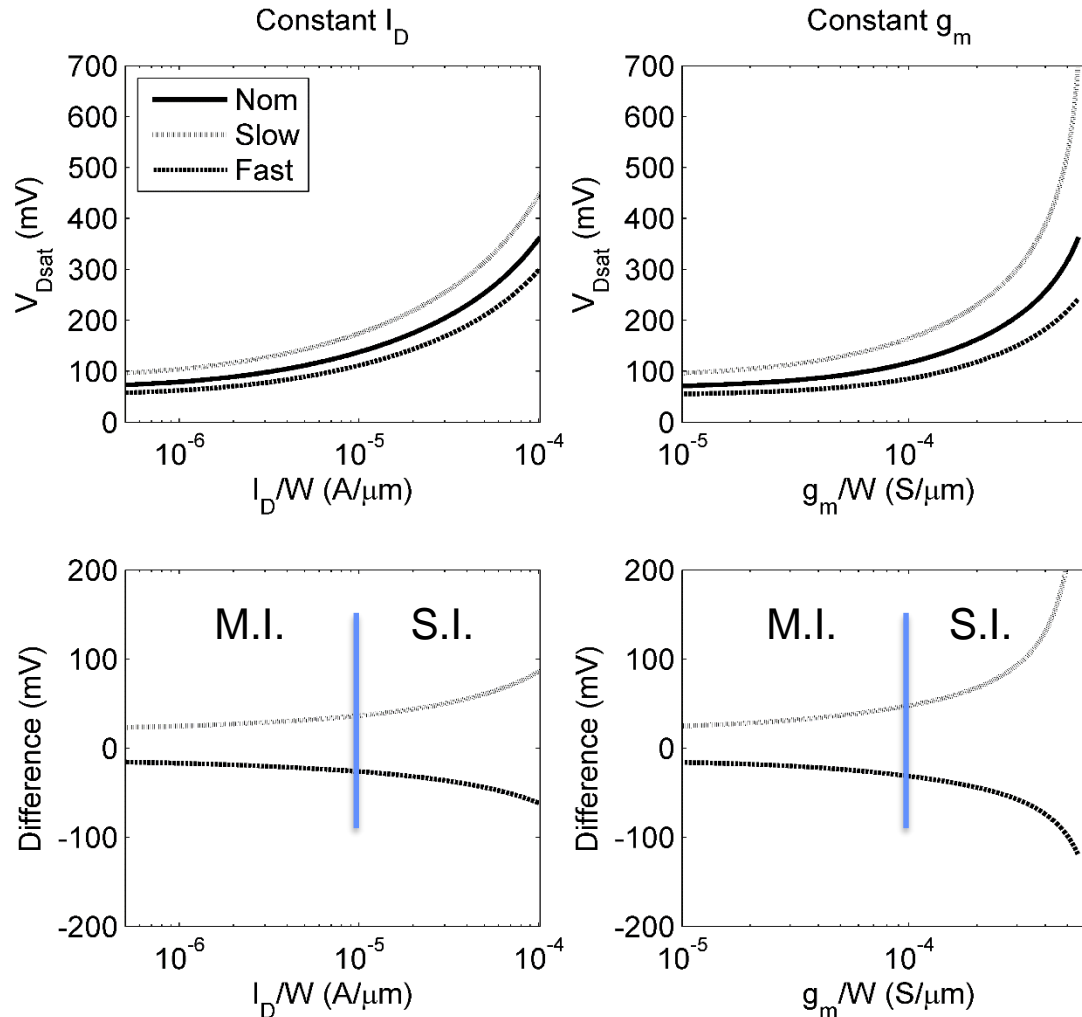
GBW varies
 like g_m/I_D
 $\sim \pm 25\%$



I_D varies like
 $(g_m/I_D)^{-1}$
 $\sim -30 \dots +60\%$

Device enters
 triode region
 (large V_{Dsat})

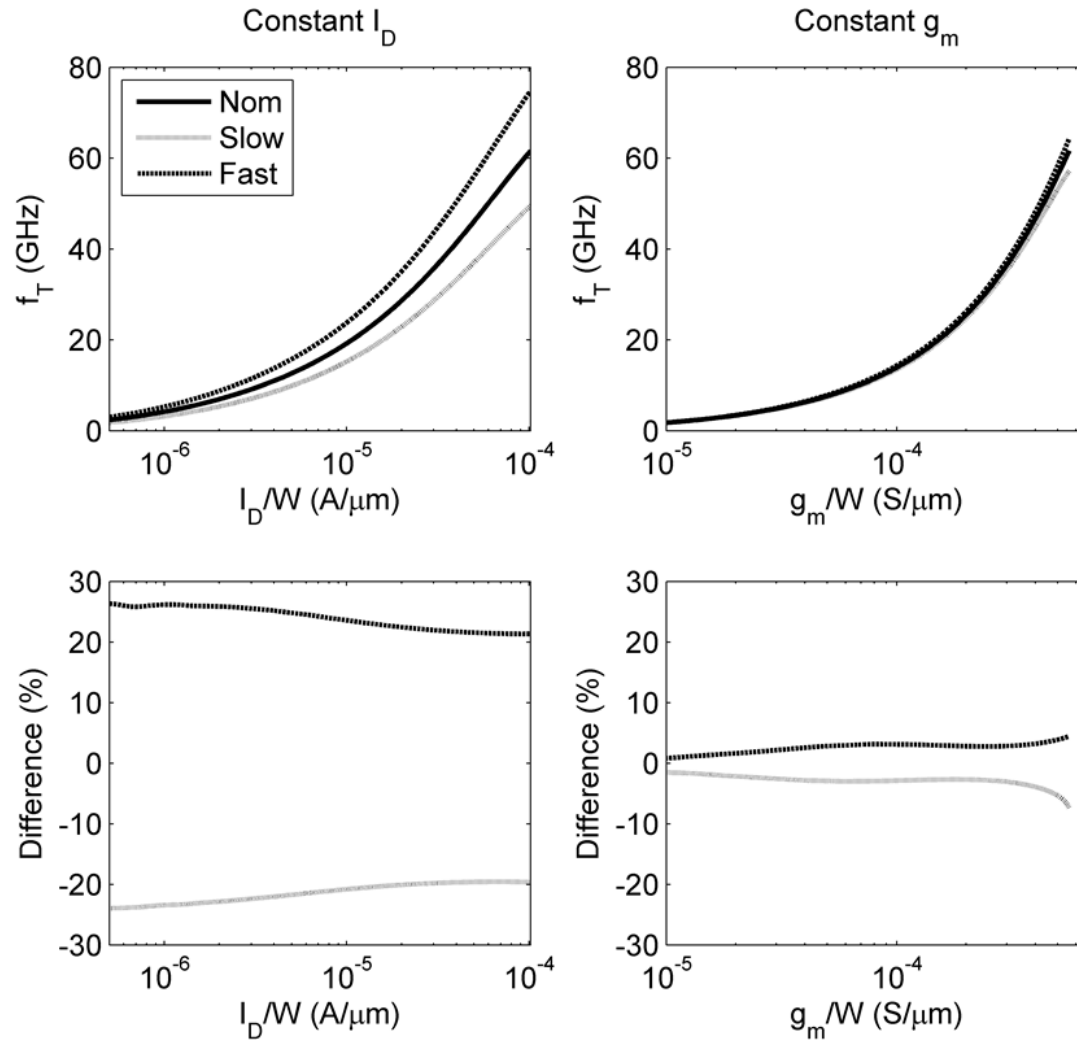
$V_{Dsat} = 2/(g_m/I_D)$ Across Corners



Strong Inversion (S.I.):
Constant g_m biasing
impractical (for low V_{DD})

Moderate Inversion (S.I.):
Constant g_m biasing OK!

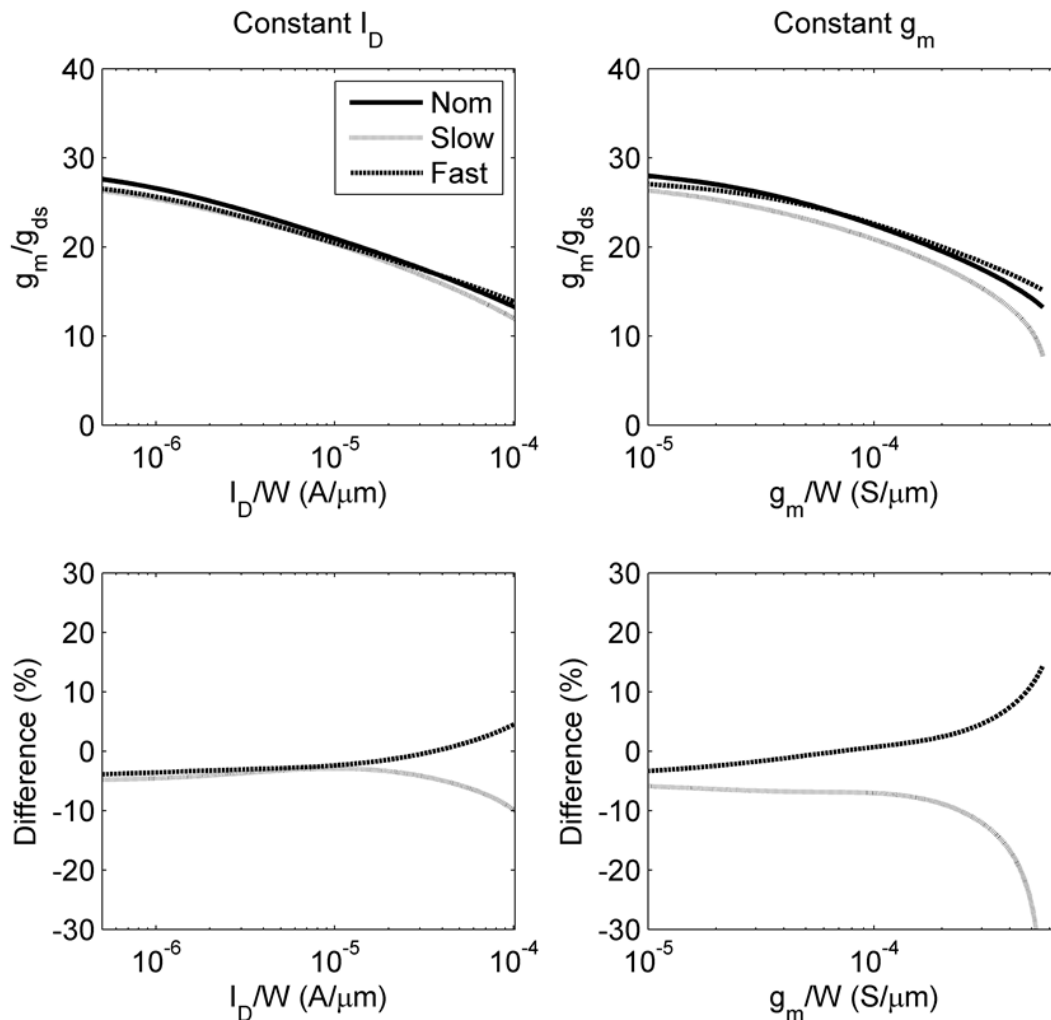
f_T Across Corners



$\sim \pm 25\%$

$\sim \pm 5\%$

g_m/g_{ds} Across Corners



Significant variations only
in strong inversion
(related to V_{Dsat} margin)

Design Flow

- Select a biasing scheme that matches your objectives
 - Constant I_D , constant g_m or something in-between
- Constant g_m design
 - Worry mostly about headroom, V_{Dsat} management (especially in strong inversion)
- Constant I_D design
 - Headroom management is typically less critical
 - Worry mostly about g_m and f_T variations ($\sim \pm 25\%$)
 - How to deal with these?

Two Options for Constant I_D Design Flow

1. Identify the “worst corner”
 - See e.g. Konishi et al., 2011
 - Design using lookup tables that represent the worst case
 - Typically slow+hot in class-A circuits (beware of exceptions!)

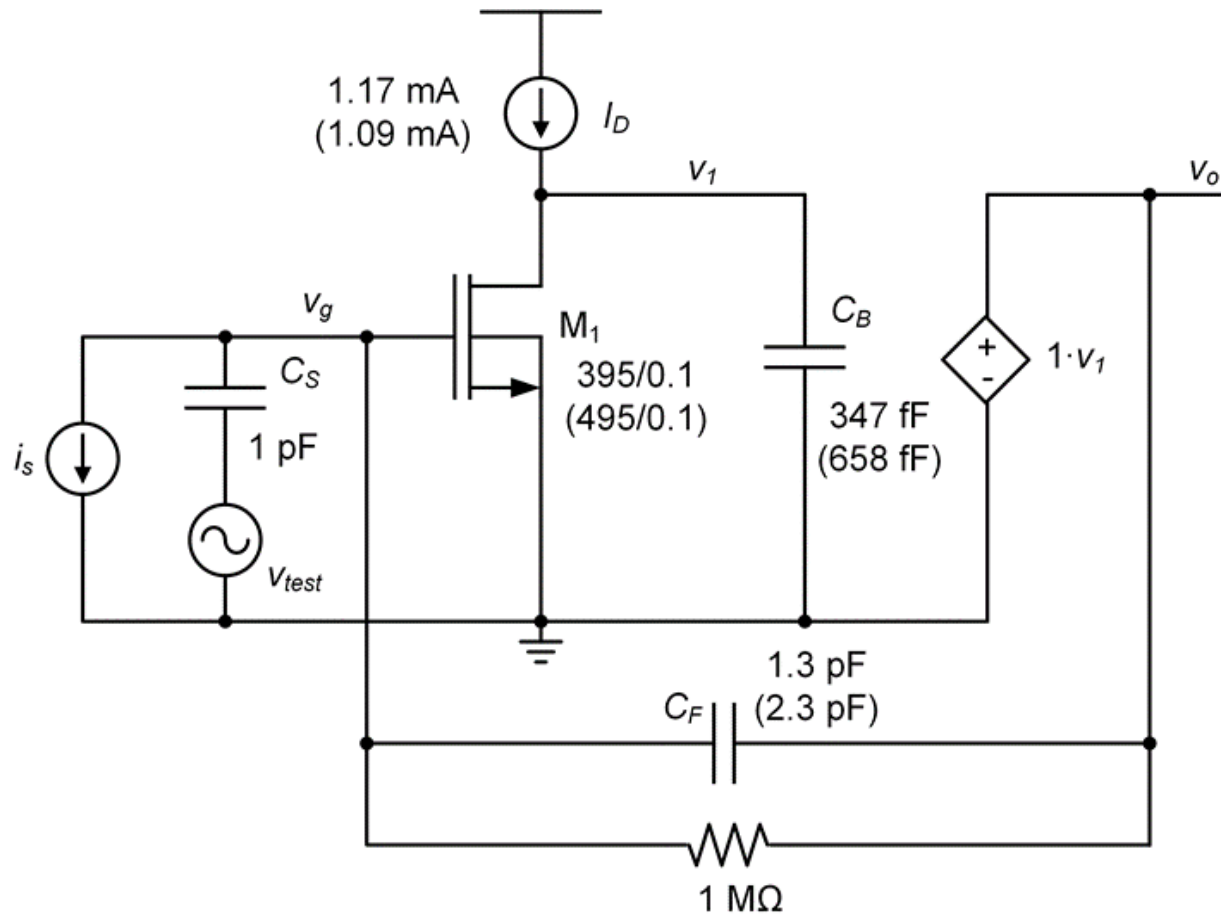
2. Pre-distort the design specifications
 - Knowing that we will lose 25% GBW in the worst corner, overdesign BW by 25% using nominal lookup table data

Required for both options: Design validation across all corners in Spice → No free lunch...

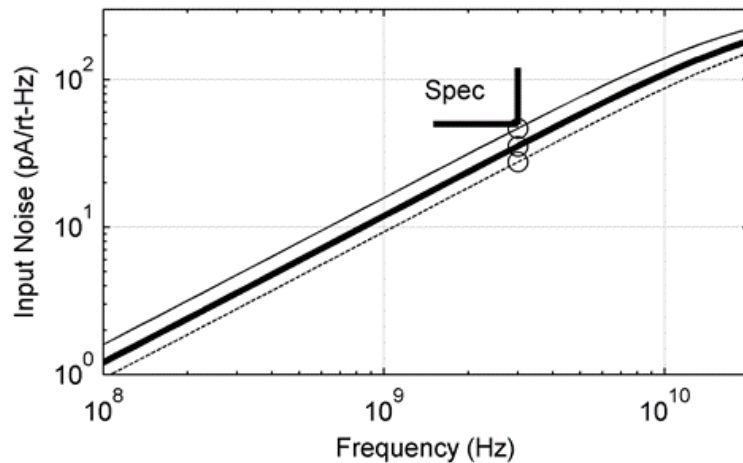
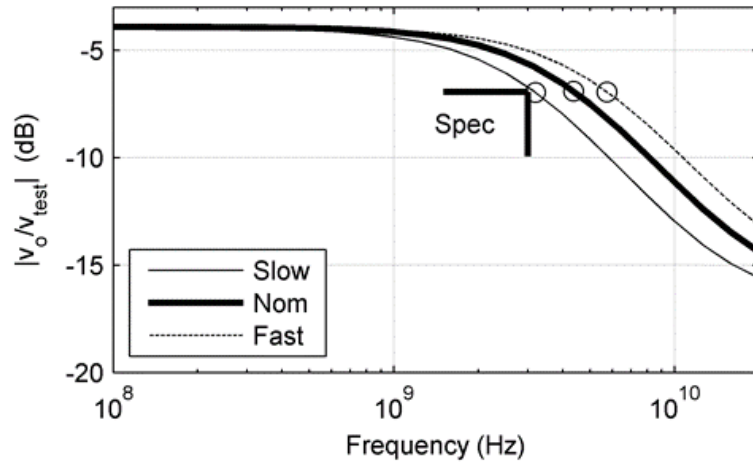
Example: Charge Amplifier Design

- Specs: $GBW = 3\text{GHz}$, Noise @ $3\text{GHz} = 50\text{ fA/rt-Hz}$, $C_s = 1\text{pF}$
- To meet the GBW and noise spec in the slow/hot corner, overdesign GBW by $\sim 1/0.7 = 1.4$
- GBW target is 4.2 GHz
- Now find optimum sizing of device as discussed earlier...

Schematic



Spice Results

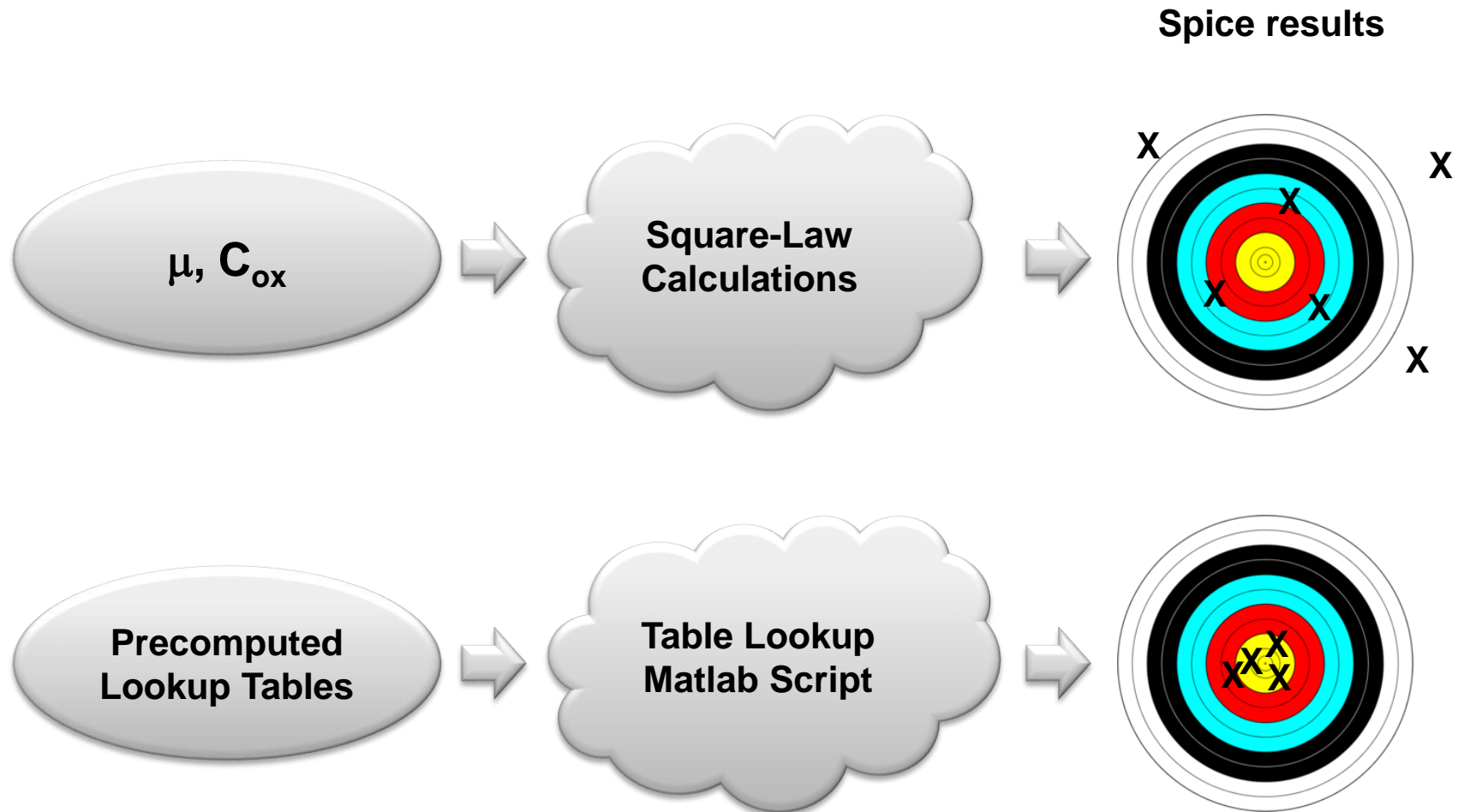


	Slow	Nom	Fast
f_c (GHz)	3.19	4.37	5.74
Deviation (%)	-27	0	+31
Noise at 3 GHz (pA/rt-Hz)	46.8	35.4	27.7
Deviation (%)	+32	0	-21

Summary

- Think g_m/I_D !
 - Weak inversion > 20 S/A \rightarrow low power design
 - Moderate inversion $10\dots 20$ S/A \rightarrow offers a nice compromise
 - Strong inversion < 10 S/A \rightarrow high-speed design
- g_m/I_D shows up naturally in many circuit calculations and provides a link between important small- and large-signal parameters
- g_m/I_D -based design using pre-computed look-tables enables
 - Systematic circuit optimization using sweeps that are hard (or impossible) to perform in Spice
 - Efficient technology node porting via script re-use
 - Sanity checking of Spice results
 - All Matlab parameters match Spice simulation closely
 - Different from square law calculations using μC_{ox} , $V_{GS}-V_t$, which have little to no significance in Spice

Comparison



References

- F. Silveira, D. Flandre, and P. G. A. Jespers, "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, vol. 31, no. 9, pp. 1314–1319, Sep. 1996.
- D. Foty, M. Bucher, D. Binkley, "Re-interpreting the MOS transistor via the inversion coefficient and the continuum of g_m/I_D ," Proc. Int. Conf. on Electronics, Circuits and Systems, pp. 1179-1182, Sep. 2002.
- P. Jespers, The g_m/I_D Methodology, a sizing tool for low-voltage analog CMOS Circuits. Springer, 2010.
- D.M. Binkley, Tradeoffs and optimization in analog CMOS design. Wiley, 2008.
- A. Shameli and P. Heydari, "A Novel Power Optimization Technique for Ultra-Low Power RFICs," in Proc. ISLPED, 2006, pp. 274–279.
- T. Taris, J. Begueret, and Y. Deval, "A 60 μ W LNA for 2.4 GHz wireless sensors network applications," in Proc. RF IC Symposium, 2011, pp. 1–4.
- T. Chan Carusone, D. A. Johns, and K. W. Martin, Analog Integrated Circuit Design, 2nd ed. Wiley, 2011.
- W. M. C. Sansen and Z. Y. Chang, "Limits of low noise performance of detector readout front ends in CMOS technology," IEEE Trans. Circuits Syst., vol. 37, no. 11, pp. 1375–1382, 1990.
- B. Murmann, "Thermal Noise in Track-and-Hold Circuits: Analysis and Simulation Techniques," IEEE Solid-State Circuits Mag., vol. 4, no. 2, pp. 46–54, 2012.
- H. C. Yang and D. J. Allstot, "Considerations for fast settling operational amplifiers," IEEE Trans. Circuits Syst., vol. 37, no. 3, pp. 326–334, Mar. 1990.
- T. Konishi, K. Inazu, J. G. Lee, M. Natsui, S. Masui, and B. Murmann, "Design Optimization of High-Speed and Low-Power Operational Transconductance Amplifier Using g_m/I_D Lookup Table Methodology," IEICE Trans. Electron., vol. E94–C, no. 3, pp. 334–345, Mar. 2011.