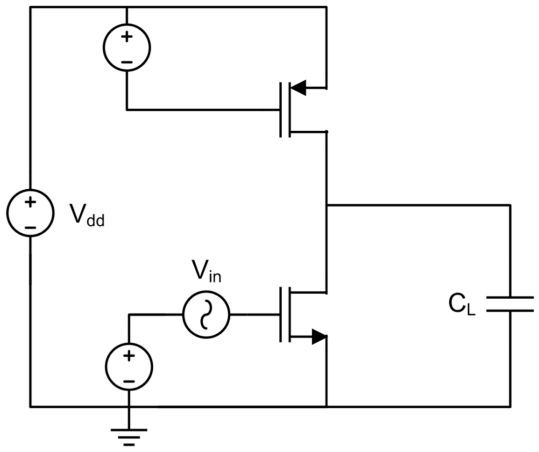
Gm over ID Methodology

Design Optimization Example (CS with active load)

source: Drew Hall (Stanford University, 2011)



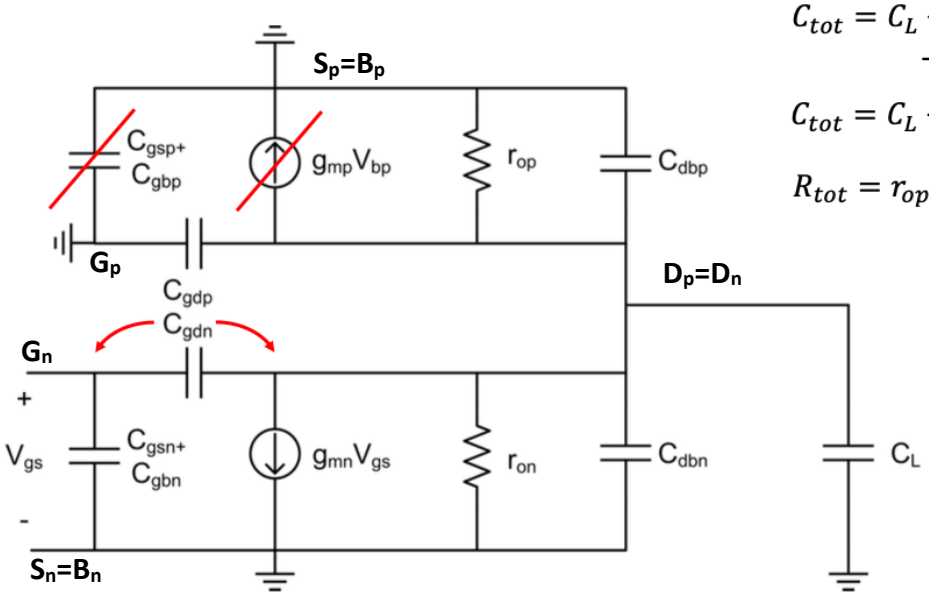
1. *Objectives*
   - Illustrate the use of the gm/Id design methodology
   - Design of a common source amplifier with a PMOS load
   - Discuss biasing without relying on a voltage source

2. *Specifications*
   - 0.18µm CMOS Technology
   - V_{DD} = 1.8V
   - A_v = 30 dB
   - P_{diss} =2 mW (neglecting bias)
   - C_L =500 fF
   - Ratio of gm/ID of NMOS to PMOS must be 2 (for noise considerations)
   - **Maximize the BW**

Disclaimer: This circuit would need feedback to properly set the output voltage. This ONLY works in simulation where we can control the offset! It is meant to illustrate the design process only.

Design – Small Signal Model

- Sketch out small signal model for transistors in the signal path
- The small signal circuit has a single pole at the output node (use Miller effect for C_{gdn})



$$C_{tot} = C_L + C_{dbn} + C_{gdn} + C_{dbp} + C_{gdp}$$

$$C_{tot} = C_L + C_{ddn} + C_{ddp}$$

$$R_{tot} = r_{op} || r_{on}$$

Design Equations

- Small signal gain:

$$|A_v(0)| = g_{mn} R_{tot}$$

- Small signal bandwidth:

$$BW = \frac{1}{2\pi} \frac{1}{R_{tot} C_{tot}}$$

- Power consumption:

$$Pwr = V_{dd} I_d$$

- Area:

$$Area = W_p L_p + W_n L_n$$

$$R_{tot} = r_{op} || r_{on}$$

Rewrite with primary gm/id design variables

$$R_{tot} = \frac{1}{\frac{1}{\frac{gmro_n}{g_{mn}}} + \frac{1}{\frac{gmro_p}{g_{mp}}}}$$

Design Flow

1) Calculate I_d from power specification
2) Calculate g_{mn} and g_{mp}
3) Lookup r_{on}
4) Calculate necessary r_{op} and find appropriate L_p
5) Lookup I_d/W and size NMOS and PMOS transistors
6) Lookup f_t and calculate intrinsic capacitances
7) Calculate extrinsic capacitances
8) Calculate BW

NOTE:

More than one design flow exists, this one comes from choosing gm/Id_n and L_n as the primary design variables

MATLAB Design script

Design_CS_PMOS_Load_Sweep.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% original script:
% Drew Hall
% EE 214 (Winter 2011)
% 18 Jan 2011
%
% Design a common source amplifier with a PMOS load using the gm/id design
% methodology.
%
% modified script:
% Claudio Talarico
% EE406 9fall 2021)
% adapted to use toolkit v.2_3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clearvars;
close all;
clear all;
clc;

format short eng;

% Constraints
Pwr = 2e-3;          % Power dissipation [Watts]
Av_dB = 30;          % Gain [dB]     (Maximum 35 with this arch)
C_L = 500e-15;       % Load capacitance [Farads]
Vdd = 1.8;           % Supply Voltage [Volts]
k = 1/2;             % Ratio of (gm/id_p) / (gm/id_n);
```

```
% Primary Design Variables
gmid_n = (5:1:24);
Ln = (0.18:0.02:0.5); % REMEBER: units of L are expressed in um

Area = zeros(length(gmid_n), length(Ln));
BW = zeros(length(gmid_n), length(Ln));

% Design a bunch of different amplifiers
for i = 1:length(gmid_n)
    for j = 1:length(Ln)
        [Area(i,j), BW(i,j)] = ...
            CS_PMOS_Load_function(Pwr, Av_dB, C_L, ...
            Vdd, k, gmid_n(i), Ln(j));
    end
end

% Plot
[X,Y] = meshgrid(Ln, gmid_n);

h = figure;
set(h,'Name','Bandwidth','NumberTitle','off')
set(h, 'Position', [100 100 900 500]);
axes('LineWidth',1.5,'FontWeight','bold','FontSize',10);
hold on;
set(gca, 'FontName','Arial Narrow', 'FontWeight', 'bold');
xlabel('L_n [\mum]', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('Gm/Id_n [S/A]', 'FontSize', 12, 'FontWeight', 'bold');
title('BW [MHz]', 'FontSize', 14, 'FontWeight', 'bold');

surf(X,Y, BW ./ 1e6);
colormap(jet);
view([0 90]);
xlim([Ln(1), Ln(end)]);
colorbar;

% Save the figure
set(gcf, 'Units', 'inches');
set(gcf, 'Position', [1 1 9 6]);
set(gcf, 'PaperPositionMode', 'auto');
print(h, '-dtiff', '-r150', sprintf('Plots_BW_%.0fdB_%.1fmW.tif', ...
    Av_dB, Pwr.*1e3));

% Find the best design
[BW_max, idx_max] = max(BW(:));
gmid_nmax = Y(idx_max);
Ln_max = X(idx_max);

[Area_max, BW_max, Wn_max, Ln_max, Wp_max, Lp_max] = ...
    CS_PMOS_Load_function(Pwr, Av_dB, C_L, Vdd, k, gmid_nmax, Ln_max);

% Print the results
disp(sprintf('\nDesign Summary of maximum BW'));
disp(sprintf('-----------------------------------'));
disp(sprintf('BW = %.3f MHz\tArea = %.3f um^2', BW_max ./1e6, Area_max));
disp(sprintf('Wn = %.3f um\t\tWp = %0.3f um', Wn_max, Wp_max));
disp(sprintf('Ln = %.3f um\t\tLp = %0.3f um', Ln_max, Lp_max));
```

CS_PMOS_Load_function.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% original script:
% Drew Hall
% EE 214 (Winter 2011)
% 18 Jan 2011
%
% Design of common source amplifier with PMOS load using the gm/id design
% methodology.
%
% modified script:
% Claudio Talarico
% EE406 (Fall 2021)
% Adapted to use toolkit v.2_3
% 08 Aug. 2021
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Area, BW, Wn, Ln, Wp, Lp] = CS_PMOS_Load_function(Pwr, Av_dB, C_L, Vdd, k,
gmid_n, Ln)

% path to lookup functions and technology files
addpath('/usr/class/gmidLUTs;/usr/class/gmidTECHs')

% Load technology files
load('180nch.mat');
load('180pch.mat');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Av = 10^(Av_dB/20);
id = Pwr/Vdd;       % Use all the available current

% Calculate the transconductances
gm_n = gmid_n*id;
gmid_p = gmid_n * k;
gm_p = gmid_p*id;

% Calculate the length of each transistor based on the gain
R = Av / gm_n;
ro_n = look_up(nch, 'GM_GDS', 'GM_ID', gmid_n, 'L', Ln) / gm_n;
ro_p_min = 1*R*ro_n/(ro_n-R);

gmgds_p_min = gm_p*ro_p_min;

gmgds_vec = look_up(pch, 'GM_GDS', 'GM_ID', gmid_p, 'L', pch.L);
M = gmgds_vec >= gmgds_p_min;

% if there is no length satisfying ro_p_min we must set Lp to NaN
if(~M)
    Lp = NaN;
else
    Lp = min(pch.L(M)); % miminum length satisfying ro_p_min
end

% Check to see if we cannot achieve this gain regardless of sizing
if(ro_p_min < 0 || isnan(Lp))
   Area = -inf; BW = -inf; Wn = 0; Ln = 0; Wp = 0; Lp = 0;
   return;
end

ro_p = look_up(pch, 'GM_GDS', 'GM_ID', gmid_p, 'L', Lp) / gm_p;
```

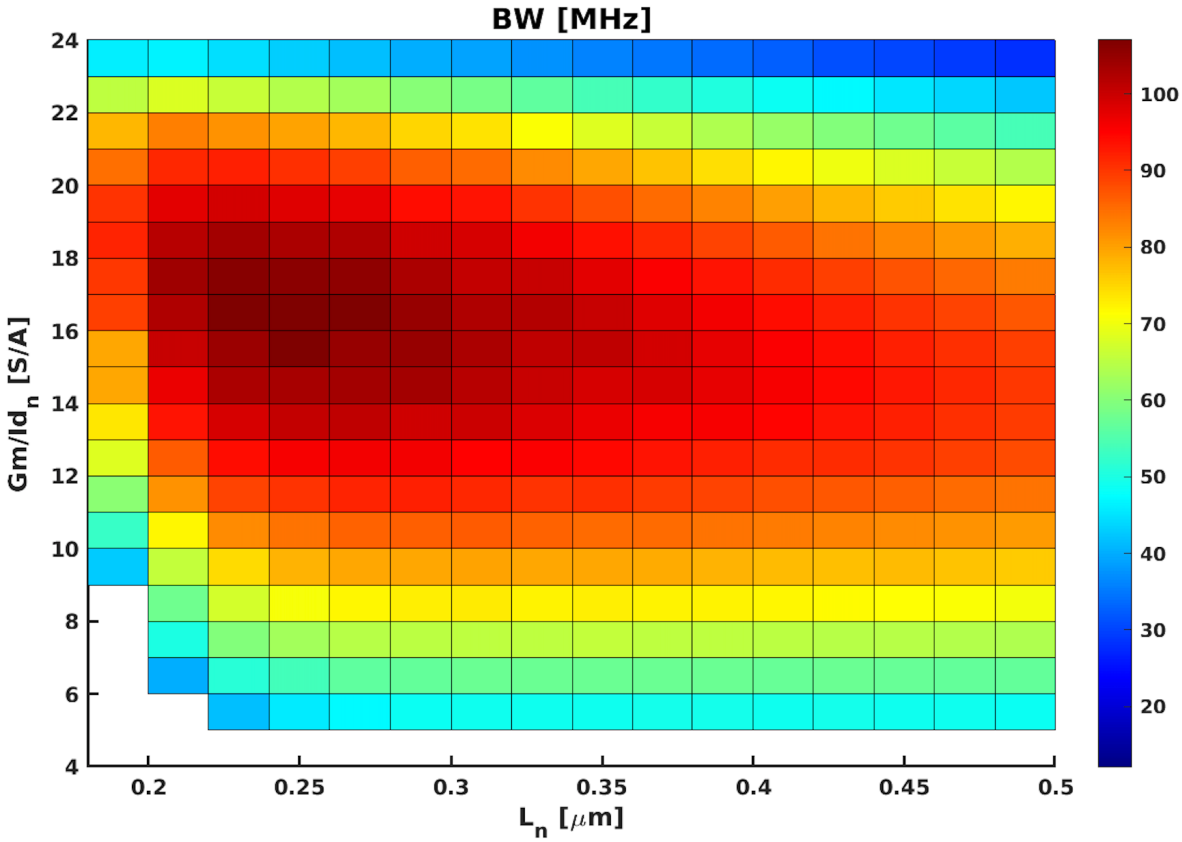
```
% Size the transistors
Wn = id / look_up(nch, 'ID_W', 'GM_ID', gmid_n, 'L', Ln);
Wp = id / look_up(pch, 'ID_W', 'GM_ID', gmid_p, 'L', Lp);

% Calculate the capacitances
wT_n = look_up(nch,'GM_CGG', 'GM_ID', gmid_n, 'L', Ln);
Cgg_n = gm_n / wT_n;
wT_p = look_up(pch, 'GM_CGG', 'GM_ID', gmid_p, 'L', Lp);
Cgg_p = gm_p / wT_p;

% since we already know the transistors width
Cdd_n = Wn*look_up(nch, 'CDD_W', 'GM_ID', gmid_n, 'L', Ln);
Cdd_p = Wp*look_up(pch, 'CDD_W', 'GM_ID', gmid_p, 'L', Lp);
Cgs_n = Wn*look_up(nch, 'CGS_W', 'GM_ID', gmid_n, 'L', Ln);
Cgs_p = Wp*look_up(pch, 'CGS_W', 'GM_ID', gmid_p, 'L', Lp);
Cgd_n = Wn*look_up(nch, 'CGD_W', 'GM_ID', gmid_n, 'L', Ln);
Cgd_p = Wp*look_up(pch, 'CGD_W', 'GM_ID', gmid_p, 'L', Lp);

% alternatively, if we didn't know the width we could have used the
% following syntax:
% This is just for sake of showing off different syntax options :-)

% Cdd_Cgg_n = look_up(nch, 'CDD_CGG', 'GM_ID', gmid_n, 'L', Ln);
% Cdd_n = Cdd_Cgg_n*Cgg_n
% Cdd_Cgg_p = look_up(pch, 'CDD_CGG', 'GM_ID', gmid_p, 'L', Lp);
% Cdd_p = Cdd_Cgg_p*Cgg_p
% Cgd_Cgg_n = look_up(nch, 'CGD_CGG', 'GM_ID', gmid_n, 'L', Ln);
% Cgd_n = Cgd_Cgg_n*Cgg_n
% Cgs_n = Cgg_n - Cgd_n
% Cgd_Cgg_p = look_up(pch, 'CGD_CGG', 'GM_ID', gmid_p, 'L', Lp);
% Cgd_p = Cgd_Cgg_p*Cgg_p
% Cgs_p = Cgg_p - Cgd_p
% Cdb_n = Cdd_n - Cgd_n
% Cdb_p = Cdd_p - Cgd_p

% Compute the total capacitance at the output node [F]
Ctot = C_L + Cdd_n + Cdd_p; % Total capacitance at output node [F]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metrics
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Area = Wn*Ln + Wp*Lp;              % [um^2]
BW = 1/(2*pi) * 1/(R*Ctot);        % [Hz]

return
```

Design Script result

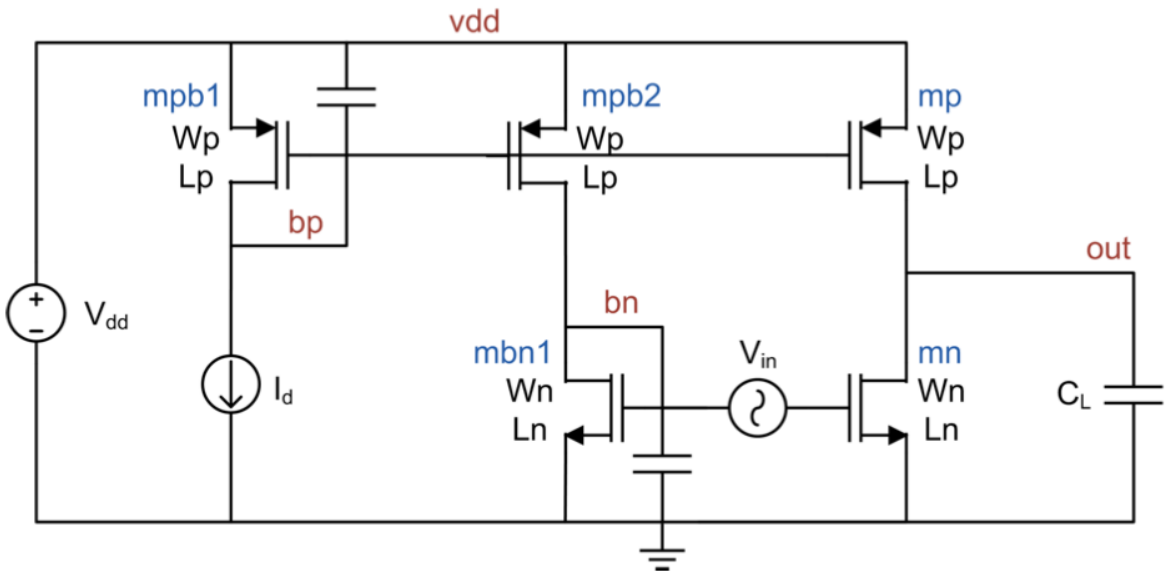
Spec: Power = 2mW, DC gain = 30dB

Design Summary of maximum BW

BW = 107.160 MHz	Area = 58.914 um^2
Wn = 147.984 um	Wp = 109.824 um
Ln = 0.220 um	Lp = 0.240 um



Circuit Design (with Biasing)



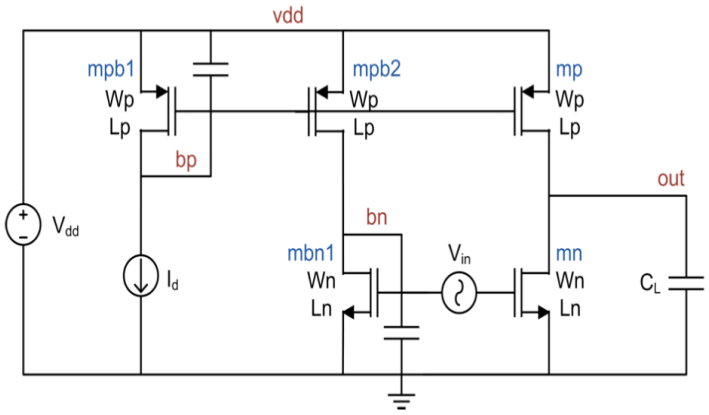
Wn = 148 µm
Ln = 0.22 µm

Wp = 110 µm
Lp = 0.24 µm

Disclaimer: This circuit would need feedback to properly set the output voltage. This ONLY works in simulation where we can control the offset! It is meant to illustrate the design process only.

SPICE Netlist

```
** file: cs_activeload.sp
*
.notrc
.option nomod noascii
.option compat          * HSPICE model and netlist compatibility
.option POST_VERSION=9601 post=1 * HSPICE database format
.option aex             * save measurements in .aex file

.include /usr/class/models/ee214_hspice.mod

.param Wp=110u Lp=0.24u Wn=148u Ln=0.22u id=1.1m

.temp 27

* circuit
mpb1 bp bp vdd vdd pch W='Wp' L='Lp'
mpb2 bn bp vdd vdd pch W='Wp' L='Lp'
mnb1 bn bn 0 0 nch W='Wn' L='Ln'
Cbp1 bp vdd 100u
Cbn1 bn 0 100u

mp out bp vdd vdd pch W='Wp' L='Lp'
mn out in 0 0 nch W='Wn' L='Ln'
CL out 0 500f

vdd vdd 0 DC 1.8
vin in bn AC 1 DC 0
ib bp 0 'id'

* Analysis
.op
.AC dec 1000 10K 10G
.PZ v(out)

* plots and measurements
.probe AC v(out)
.defwave v_out=vm(out)
.extract ac label=gain_db max(vdb(out))
.extract ac label=f3db 'crossing(vdb(out), gain_db-3)'

.end
```



.OP Simulation (vs. calculated)

Simulated **Calculated** **%Error**

	MPB1	MPB2	MNB1	MP	MN		
MODEL	PCH	PCH	NCH	PCH	NCH		
ID	-1.1000E-03	-1.2122E-03	1.2122E-03	-1.2122E-03	1.2122E-03	1.1mA, 1.1mA	+10.2%
VGS	-7.0462E-01	-7.0462E-01	5.9125E-01	-7.0462E-01	5.9125E-01		
VDS	-7.0462E-01	-1.2087E+00	5.9125E-01	-1.2087E+00	5.9125E-01		
VTH	-4.7330E-01	-4.7110E-01	4.8196E-01	-4.7110E-01	4.8196E-01		
VDSAT	-2.0134E-01	-2.0303E-01	9.7884E-02	-2.0303E-01	9.7884E-02		
GM	8.7809E-03	9.3811E-03	1.8861E-02	9.3811E-03	1.8861E-02	8.9mS, 17.8mS	+ 5.9%
GDS	2.5377E-04	2.0211E-04	3.9114E-04	2.0211E-04	3.9114E-04	213.53uS, 347.63uS	+12.5%
Cdd	1.5492E-13	1.4621E-13	1.7900E-13	1.4621E-13	1.7900E-13	155.07fF, 180.16fF,	–5.71%
Cgg	2.8449E-13	2.8445E-13	3.3935E-13	2.8445E-13	3.3935E-13		
Cgd	-7.1863E-14	-7.1833E-14	-7.1576E-14	-7.1833E-14	-7.1576E-14		
Cgs	-2.0556E-13	-2.0538E-13	-2.4911E-13	-2.0538E-13	-2.4911E-13		
Region	Saturation	saturation	Saturation	saturation	saturation		
VTH_D	2.3133E-01	2.3353E-01	1.0930E-01	2.3353E-01	1.0930E-01		

.PZ Simulation (vs. calculated)

Simulated **Calculated** **%Error**

```
POLES          REAL PART          IMAGINARY PART
    1          -1.137972e+08      -0.000000e+00
ZEROS          REAL PART          IMAGINARY PART
    1           4.089441e+10      -0.000000e+00
```

106.93MHz **+6.42%**

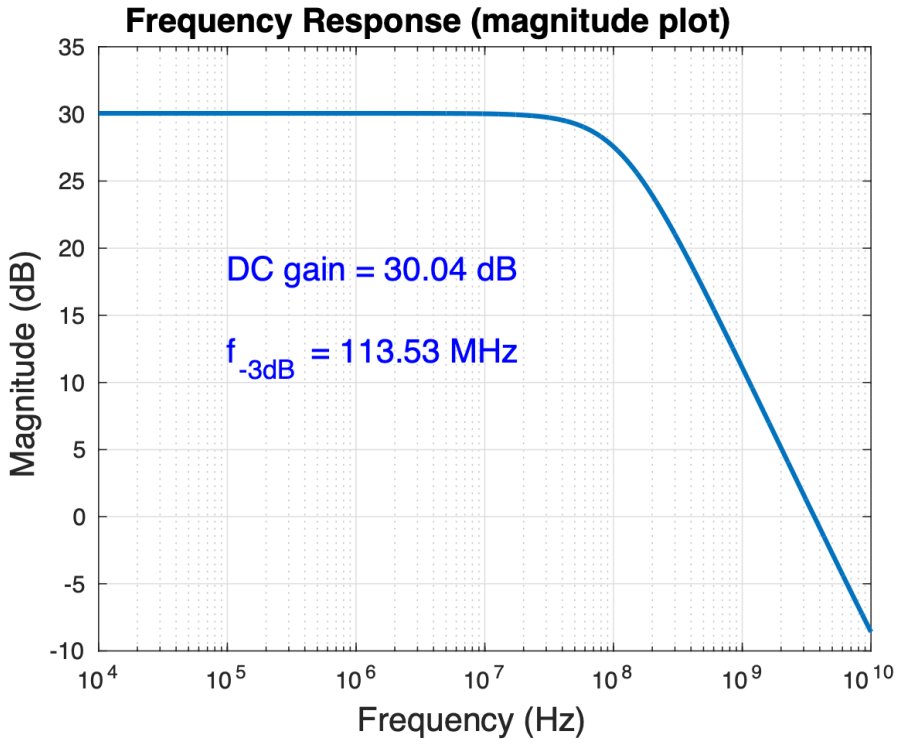
.AC Simulation (vs. calculated)

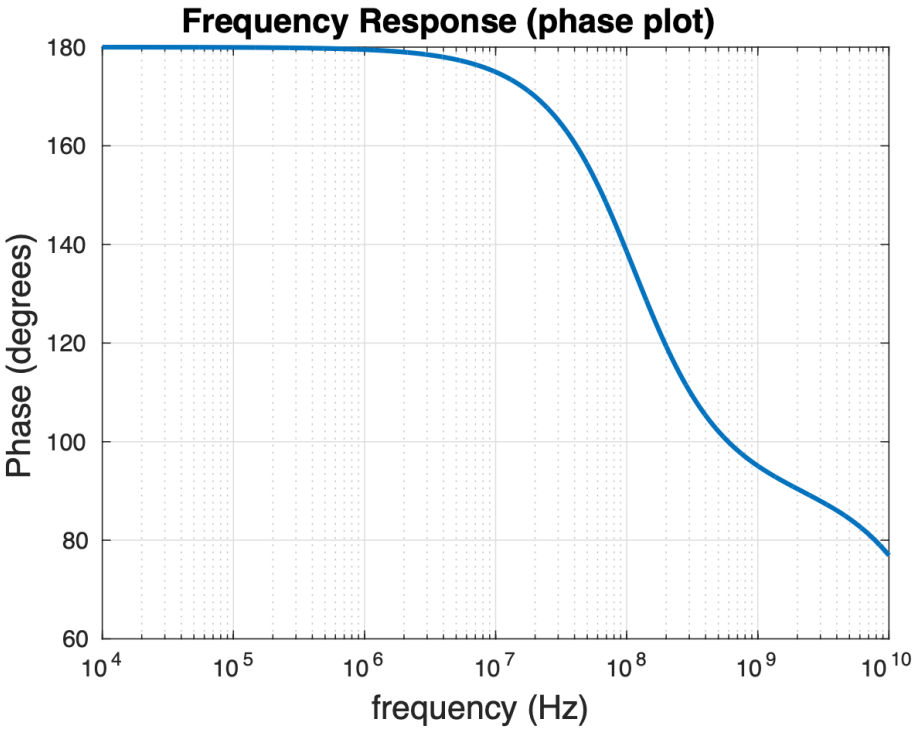
Simulated **Calculated** **%Error**

```
TEMPERATURE =  2.7000E+01 Celsius
gain_db =  3.0042E+01
f3db =  1.1353E+08
```

30 dB **+0.14%**

106.93MHz **+6.17%**


Frequency Response (magnitude plot)
DC gain = 30.04 dB
f -3dB = 113.53 MHz
Magnitude (dB)
Frequency (Hz)

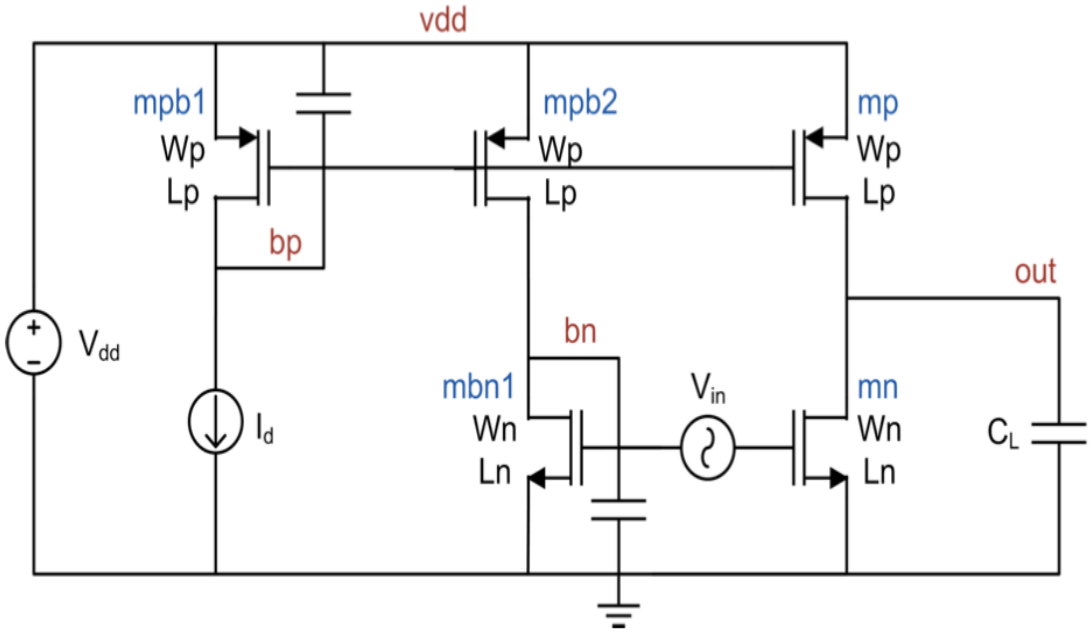

Frequency Response (phase plot)
Phase (degrees)
frequency (Hz)

Observations

- Using the gm/id design methodology, we can easily hit our design target without resorting to spice monkeying

- If we want, we can resolve the small (~10%) discrepancies through "educated" tweaking
    - i.e, we know what knobs to adjust and what to expect, we do not do random tweaking (monkeying!)
    - the simulated current was about 10.2% higher than expected. This resulted in our g_m also being slightly higher. Ultimately, this happened because of the V_{ds} dependence in our current mirror. Mbp1 had exactly 1.1 mA whereas mbp2 had 1.21mA.

	MPB1	MPB2	MNB1	MP	MN
MODEL	PCH	PCH	NCH	PCH	NCH
ID	-1.1000E-03	-1.2122E-03	1.2122E-03	-1.2122E-03	1.2122E-03
VGS	-7.0462E-01	-7.0462E-01	5.9125E-01	-7.0462E-01	5.9125E-01
VDS	-7.0462E-01	-1.2087E+00	5.9125E-01	-1.2087E+00	5.9125E-01
Region	Saturation	saturation	Saturation	saturation	saturation

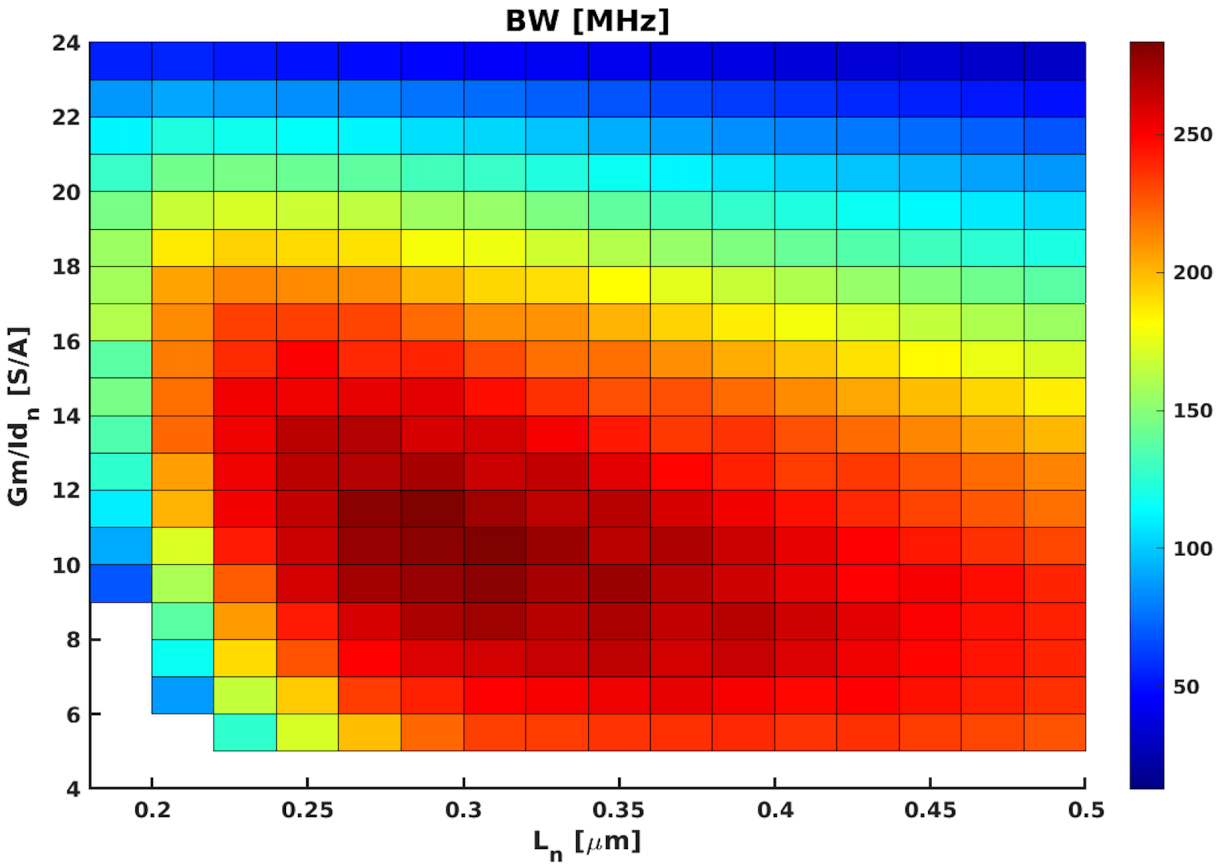


Design Space Exploration:
increased power budget to 20 mW and left gain unchanged to 30 dB

Spec: Power = 20mW, DC gain = 30dB

Design Summary of maximum BW

BW = 283.757 MHz Area = 303.344 um^2
Wn = 671.476 um Wp = 463.188 um
Ln = 0.300 um Lp = 0.220 um

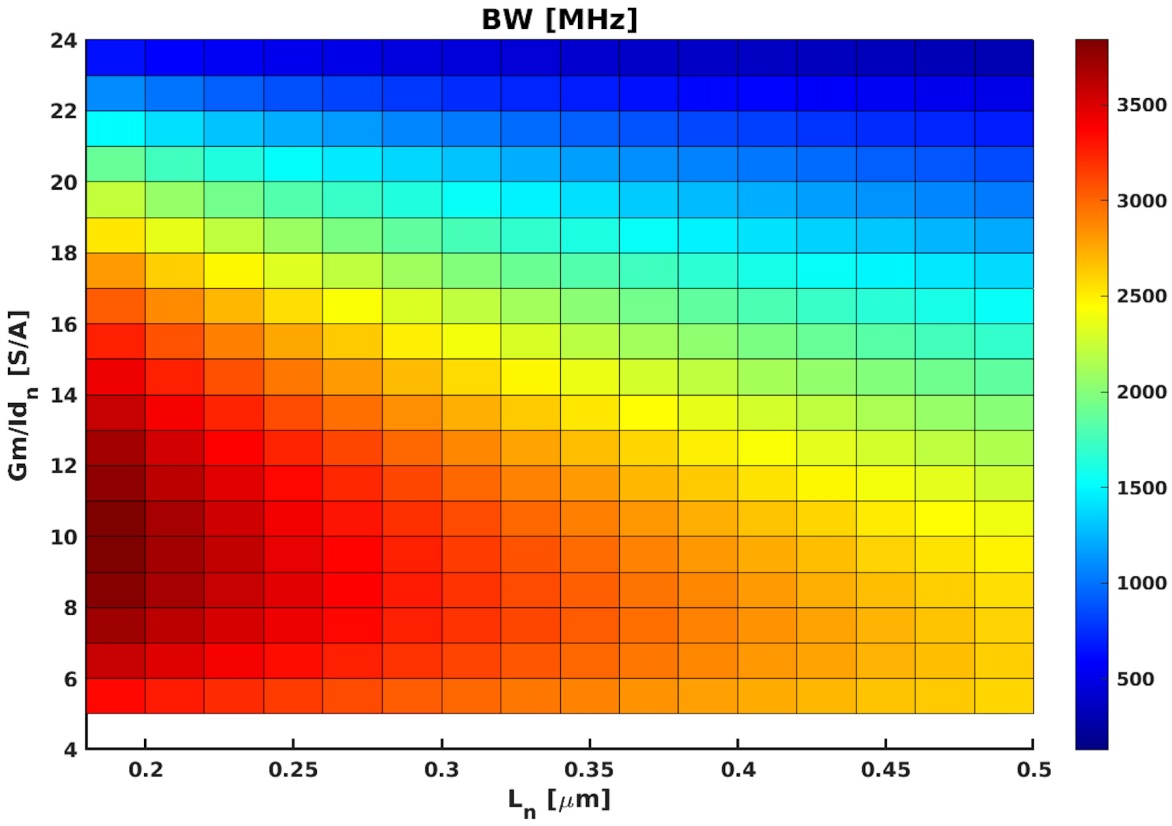


Design Space Exploration:
increased power budget to 20 mW and reduced gain to 10 dB

Spec: Power = 20mW, DC gain = 10dB

Design Summary of maximum BW

BW = 3844.287 MHz Area = 111.139 um^2
Wn = 318.590 um Wp = 298.850 um
Ln = 0.180 um Lp = 0.180 um

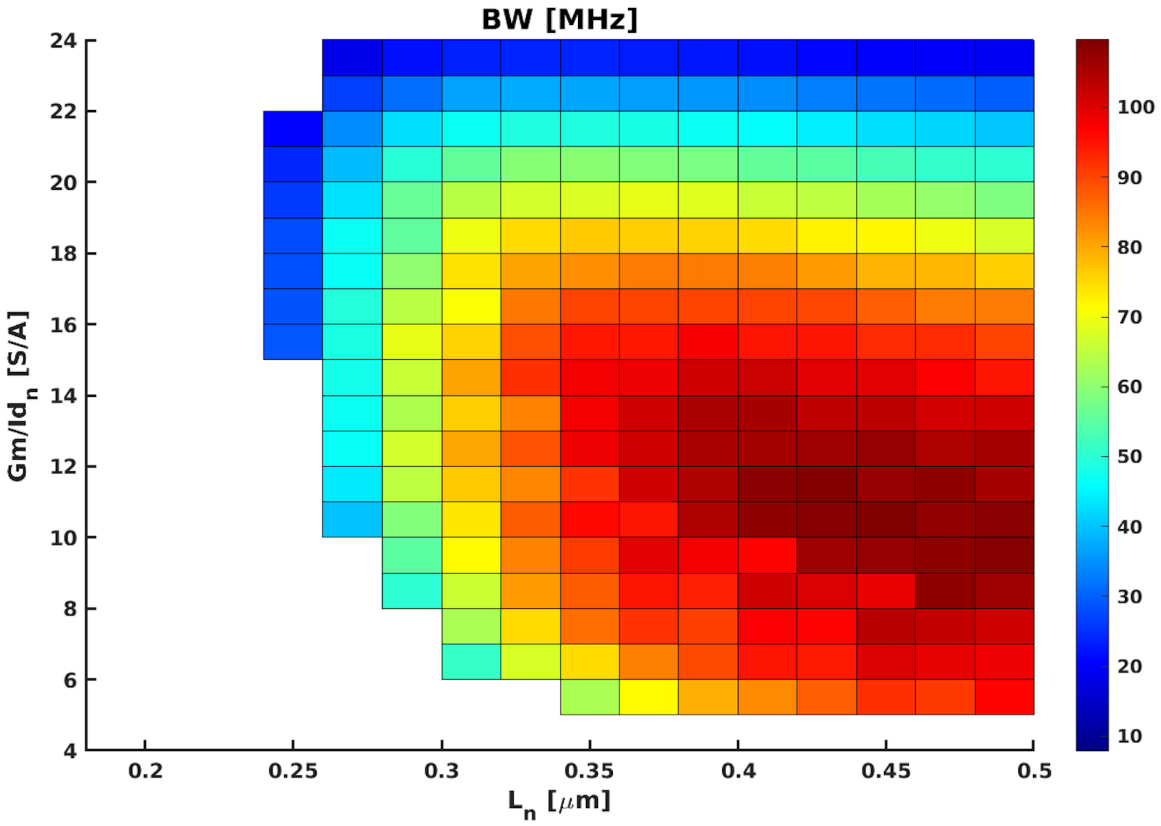


Design Space Exploration:
increased power budget to 20 mW and increased gain to 34 dB

Spec: Power = 20mW, DC gain = 34dB

Design Summary of maximum BW

BW = 109.743 MHz Area = 886.469 um^2
Wn = 1011.829 um Wp = 1050.628 um
Ln = 0.440 um Lp = 0.420 um



Design of Common Source with PMOS Load– Summary

- Using the precomputed lookup tables and the gm/id design methodology, we created an optimization script for a common source with a PMOS load
- In this example, we optimized for maximum BW, but we could easily have optimized for any other parameter (area, power, some weighted combination, etc.)
- gm/id is a powerful design tool that serves as an enabler for deep submicron processes where short channel effects make the square law equation useless for design
- Note: It is important to realize the limitations of the gm/id design methodology! It assumes a small signal model of the device (ie: it assumes the devices can be linearized around some operating point). It has limitations when designing mixers, power amplifiers, switches, etc.

MATLAB script for design analysis and simulation plots

```
% C. Talarico
% File: cs_analysis_and_plots.m

clearvars; close all; clc;

%--------------------%
% Simulation Results %
%--------------------%
addpath('/usr/local/MATLAB/HspiceToolbox');
x = loadsig('./spiceout_cs_activeload/cs_activeload.ac0')
lssig(x)

freq = evalsig(x,'HERTZ');
vout = evalsig(x,'v_out');
mag = abs(vout);
magdb = 20*log10(mag);
phase = (180/pi)*unwrap(angle(vout));

gain = max(magdb)
f3db = 1e-6*interp1(magdb, freq, magdb(1)-3, 'spline')

% Plots
figure(1);
semilogx(freq,magdb,'linewidth',2);
grid;
```

```
xlabel('Frequency (Hz)', 'fontsize',14);
ylabel('Magnitude (dB)', 'fontsize', 14);
title('Frequency Response (magnitude plot)', 'fontsize', 14)
% annotate the plot
str1 = sprintf('DC gain = %.2f dB\n', gain);
str2 = sprintf('f_{-3dB} = %.2f MHz', f3db);
str = {str1, str2};
text(1e5,15, str, 'color', 'b', 'fontsize', 14);

figure(2)
semilogx(freq, phase, 'linewidth',2);
grid;
xlabel('frequency (Hz)','fontsize',14);
ylabel('Phase (degrees)','fontsize',14);
title('Frequency Response (phase plot)','fontsize',14);

print(figure(1), '-dpsc', 'Plot_CS_activeload.ps');
print(figure(2), '-append', '-dpsc', 'Plot_CS_activeload.ps');

%--------------------%
%    Calculations    %
%--------------------%
addpath('/usr/class/gmidLUTs','/usr/class/gmidTECHs');
load('180nch.mat');
load ('180pch.mat');

Pwr = 2e-3;          % Power dissipation [Watts]
Av_dB = 30;          % Gain [dB]     (Maximum 35 with this arch)
C_L = 500e-15;       % Load capacitance [Farads]
Vdd = 1.8;           % Supply Voltage [Volts]

Ln = 0.220           % um
Wn = 148
Lp = 0.240
Wp = 110

Id = Pwr/Vdd
Av = 10^(Av_dB/20)
JD_n = Id/Wn;
JD_p = Id/Wp;
gmid_n = look_up(nch, 'GM_ID', 'ID_W', JD_n, 'L', Ln)
gm_n = gmid_n*Id
gmid_p = look_up(pch, 'GM_ID', 'ID_W', JD_p, 'L', Lp)
gm_p = gmid_p*Id
gmgds_n = look_up(nch, 'GM_GDS', 'GM_ID', gmid_n, 'L', Ln)
gds_n = gm_n/gmgds_n
gmgds_p = look_up(pch, 'GM_GDS', 'GM_ID', gmid_p, 'L', Lp)
gds_p = gm_p/gmgds_p

Rtot = 1/(gds_n + gds_p)
Cdd_n = Wn*look_up(nch,'CDD_W', 'GM_ID', gmid_n, 'L', Ln)
Cdd_p = Wp*look_up(pch,'CDD_W', 'GM_ID', gmid_p, 'L', Lp)
Ctot = C_L + Cdd_n + Cdd_p

BW = 1/(2*pi*Rtot*Ctot)
```

```
Simulated Results
-----------------------
gain = 30.0416      (dB)
f3db = 113.5282    (MHz)

Calculated Results
------------------------
Ln = 0.2200          (um)
Wn = 148             (um)
Lp = 0.2400          (um)
Wp = 110             (um)
Id = 0.0011          (A)
Av = 31.6228         (V/V)
gmid_n = 16.0028     (S/A)
gm_n = 0.0178        (S)
gmid_p = 8.0083      (S/A)
gm_p = 0.0089        (S)
gmgds_n = 51.1487
gds_n = 3.4763e-04   (S)
gmgds_p = 41.6721
gds_p = 2.1353e-04   (S)
Rtot = 1.7820e+03    (ohm)
Cdd_n = 1.8016e-13   (Farad)
Cdd_p = 1.5507e-13   (Farad)
Ctot = 8.3523e-13    (Farad)
BW = 1.0693e+08      (Hz)
```

A faster MATLAB implementation of the Design script

The “original” MATLAB implementation of the design script (`Design_CS_PMOS_Load_Sweep.m` + `CS_PMOS_Load_function.m`) takes about 913 seconds of run time.

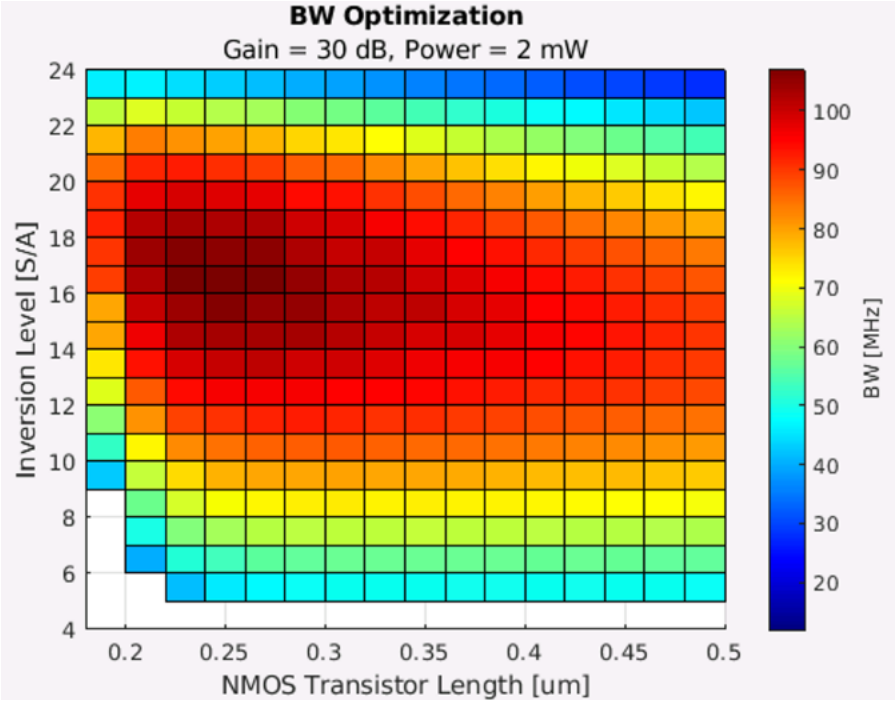
“Recoding” the script allows to reduce the run time to about 26 seconds (about 35x):

cs_activeLoad_pm.m

```
% gm/id design of Common Source with PMOS Load
% Patrick Munar
% adapted from Drew Hall (Stanford)
% Gonzaga University
% July 30th, 2021
% file: cs_activeLoad_pm.m

tic

clearvars;
close all;

addpath('/usr/class/gmidLUTs;/usr/class/gmidTECHs');

% Load transistor data
load 180nch.mat;
load 180pch.mat;

nmos = nch;
pmos = pch;

% Design Specifications
VDD = 1.8;              % Voltage source [V]
AV0_dB = 30;            % Gain in [dB]
AV0 = 10^(AV0_dB/20);   % Gain in [V/V]
CL = 500*1e-15;         % Load capacitance [fF]
P = 2*1e-3;             % Power dissipated [mW]
Ratio = 1/2;            % ratio between gm_IDp : gm_IDn

% Design Choices
Ln = [0.18:0.02:0.5];   % NMOS transistor length [um]
gm_IDn = [5:1:24];      % NMOS inversion levels [S/A]

% Assumptions and Equations:
% Use small-signal model to derive expressions for BW, Gain, and total
% resistance/capacitance

% Calculating remaining component values based on assumptions and equations
ID = P/VDD;    % Drain current [mA]

% Pre-allocation of variables for efficiency
Area            =   zeros(length(gm_IDn),length(Ln));
BW              =   zeros(length(gm_IDn),length(Ln));
Lp              =   zeros(1,length(Ln));
```

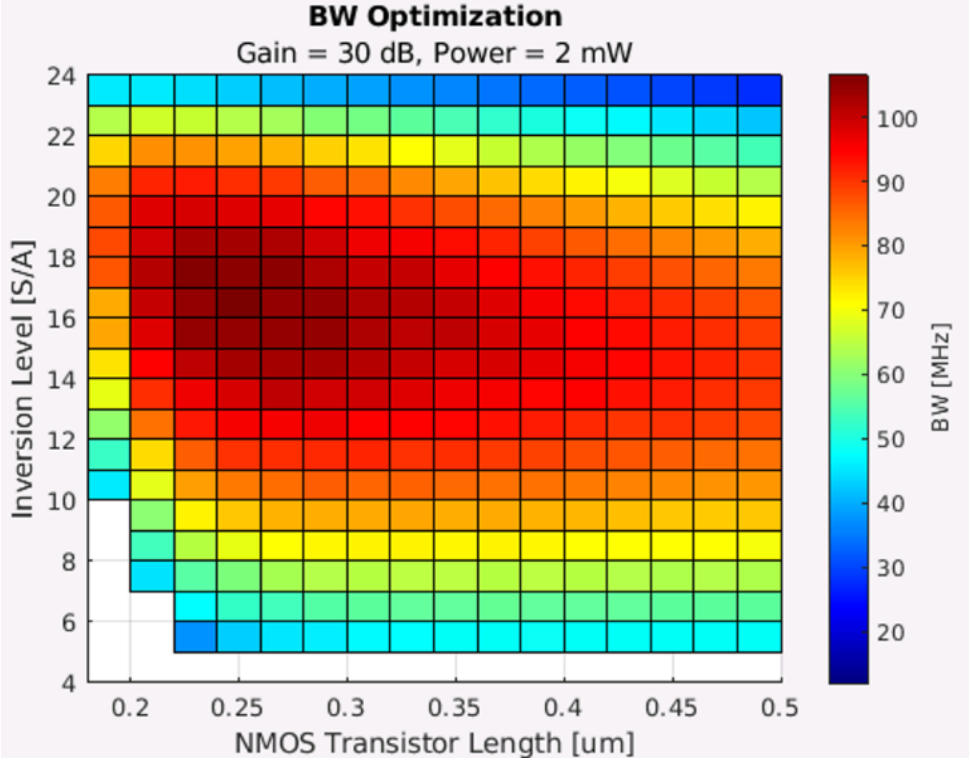
```
Wn              =   zeros(length(gm_IDn),length(Ln));
Wp              =   zeros(length(gm_IDn),length(Ln));
BW_max          =   0;

% Determining other parameters via lookup tables
for i = 1:length(gm_IDn)
  for j = 1:length(Ln)
    % Determine NMOS and PMOS transconductance and inversion levels
    gmn = gm_IDn(i)*ID;         % NMOS transconductance [S]
    gm_IDp = gm_IDn(i)*Ratio;   % PMOS inversion levels [S/A]
    gmp = gm_IDp*ID;            % PMOS transconductance [S]
    RT = AV0/gmn;               % Total resistance [kohms]

    % Determine minimum and sample PMOS intrinsic gain
    ron = look_up(nmos,'GM_GDS','GM_ID',gm_IDn(i),'L',Ln(j))/gmn;
    % NMOS output resistance [kohms]
    rop_min = (RT*ron)/(ron-RT);
    % PMOS output resistance [kohms]
    gmrop_min = gmp*rop_min;
    % PMOS intrinsic gain
    gmrop_ref = look_up(pmos,'GM_GDS','GM_ID',gm_IDp,'L',pmos.L);
    % Reference PMOS intrinsic gain

    % Checks if the samples are greater than or equal to the minimum
    % PMOS intrinsic gain
    gmrop_test = gmrop_min <= gmrop_ref;

    % Assigns appropriate PMOS length
    for k = 1:length(gmrop_test)
      if (gmrop_test(k) == true)
        Lp(j) = min(pch.L(k));
        break
      else
        Lp(j) = NaN;
      end
    end

    % Checks if it is impossible to properly size the PMOS transistor
    if (rop_min < 0)||(isnan(Lp(j)))
      Area(i,j) = -inf;
      BW(i,j) = -inf;
      Wn(i,j) = 0;
      Wp(i,j) = 0;
      Lp(j) = 0;
      continue
     end

     rop = look_up(pmos,'GM_GDS','GM_ID',gm_IDp,'L',Lp(j))/gmp;
     %PMOS output resistance [kohms]

     % Size NMOS and PMOS transistors
     Wn(i,j) = ID/look_up(nmos,'ID_W','GM_ID',gm_IDn(i),'L',Ln(j));
     % NMOS transistor width [um]
     Wp(i,j) = ID/look_up(pmos,'ID_W','GM_ID',gm_IDp,'L',Lp(j));
     % PMOS transistor width [um]
```

```
    % Calculate internal capacitances
    CGGn = gmn/look_up(nmos,'GM_CGG','GM_ID',gm_IDn(i),'L',Ln(j));
    % Total NMOS gate capacitance
    CDDn = CGGn*look_up(nmos,'CDD_CGG','GM_ID',gm_IDn(i),'L',Ln(j));
    % Total NMOS drain capacitance
    CGGp = gmp/look_up(pmos,'GM_CGG','GM_ID',gm_IDp,'L',Lp(j));
    % Total PMOS gate capacitance
    CDDp = CGGp*look_up(pmos,'CDD_CGG','GM_ID',gm_IDp,'L',Lp(j));
    % Total PMOS drain capacitance
    CT = CL + CDDn + CDDp;
    % Capacitance at the output node

    % Calculate total transistor area and bandwidth
    Area(i,j) = Wn(i,j)*Ln(j) + Wp(i,j)*Lp(j);  % Transistor area [um]^2
    BW(i,j) = 1/(2*pi*RT*CT);                   % Bandwidth [MHz]

    % Determine optimum amplifier design
    BW_test = BW(i,j);
    if (BW_test > BW_max)  % Updates variables if previous BW is exceeded
      BW_max = BW_test;
      Area_max = Area(i,j);
      Ln_max = Ln(j);
      Lp_max = Lp(j);
      Wn_max = Wn(i,j);
      Wp_max = Wp(i,j);
    end
  end
end

% Plot results
specifications = "Gain = " + AV0_dB + " dB, Power = " + P*1e3 + " mW";
figure('Name','CS Amplifier with PMOS Load','NumberTitle','off');
[X,Y] = meshgrid(Ln, gm_IDn);
surf(X,Y,BW*1e-6);
xlabel('NMOS Transistor Length [um]');
ylabel('Inversion Level [S/A]');
title('BW Optimization',specifications);
c = colorbar; c.Label.String = 'BW [MHz]';
colormap(jet);
view([0 90]);
xlim([Ln(1),Ln(end)]);

% Displaying results
fprintf('\nBW Optimization for CS Amplifier with PMOS Load\n')
fprintf('-------------------------------------------------\n');
fprintf('Maximum Bandwidth:            %3.2f [MHz]\n',BW_max*1e-6);
fprintf('Optimum Amplifier Area:       %3.2f [um^2]\n',Area_max);
fprintf('-------------------------------------------------\n');
fprintf('NMOS Dimensions (Ln x Wn):    %1.3f [um] x %3.2f
[um]\n',Ln_max,Wn_max);
fprintf('PMOS Dimensions (Lp x Wp):    %1.3f [um] x %3.2f
[um]\n',Lp_max,Wp_max);

toc
```




```
BW Optimization for CS Amplifier with PMOS Load
-------------------------------------------------
Maximum Bandwidth:            107.16 [MHz]
Optimum Amplifier Area:       58.91 [um^2]
-------------------------------------------------
NMOS Dimensions (Ln x Wn):    0.220 [um] x 147.98 [um]
PMOS Dimensions (Lp x Wp):    0.240 [um] x 109.82 [um]
Elapsed time is 26.273401 seconds.
```

If instead of using the technology files generated using HSPICE (180nch.mat and 180pch.mat) we use the technology files generated using ELDO (180nch_e.mat, 180pch_e.mat) the results obtained are slightly different (the % error on the estimated BW is about 0.4%)




```
BW Optimization for CS Amplifier with PMOS Load
-------------------------------------------------
Maximum Bandwidth:            106.75 [MHz]
Optimum Amplifier Area:       60.98 [um^2]
-------------------------------------------------
NMOS Dimensions (Ln x Wn):    0.240 [um] x 163.35 [um]
PMOS Dimensions (Lp x Wp):    0.220 [um] x 98.97 [um]
Elapsed time is 23.705611 seconds.
```

.OP Simulations

HSPICE TECH FILES | ELDO TECH FILES

	MP	MN	MP	MN
	Lp=0.24, Wp=110u	Ln=0.22u, Wp=148u	Lp=0.22u, Wp=99u	Ln=0.24u, Wn=163u
MODEL	PCH	NCH	PCH	NCH
ID	-1.2122E-03	1.2122E-03	-1.2244E-03	1.2244E-03
VGS	-7.0462E-01	5.9125E-01	-7.0982E-01	5.9109E-01
VDS	-1.2087E+00	5.9125E-01	-1.2089E+00	5.9109E-01
VTH	-4.7110E-01	4.8196E-01	-4.7677E-01	4.7836E-01
VDSAT	-2.0303E-01	9.7884E-02	-2.0297E-01	9.9923E-02
GM	9.3811E-03	1.8861E-02	9.4551E-03	1.8900E-02
GDS	2.0211E-04	3.9114E-04	2.2750E-04	3.5229E-04
Cdd	1.4621E-13	1.7900E-13	1.3166E-13	1.9725E-13
Cgg	2.8445E-13	3.3935E-13	2.4220E-13	3.9661E-13
Region	saturation	saturation	saturation	saturation
VTH_D	2.3353E-01	1.0930E-01	2.3305E-01	1.1273E-01

.AC Simulations

HSPICE TECH FILES | ELDO TECH FILES

TEMPERATURE	27 Celsius	27 Celsius
gain_db	3.00416E+01	3.0265E+01
f3db	1.135282E+08	1.1105E+08