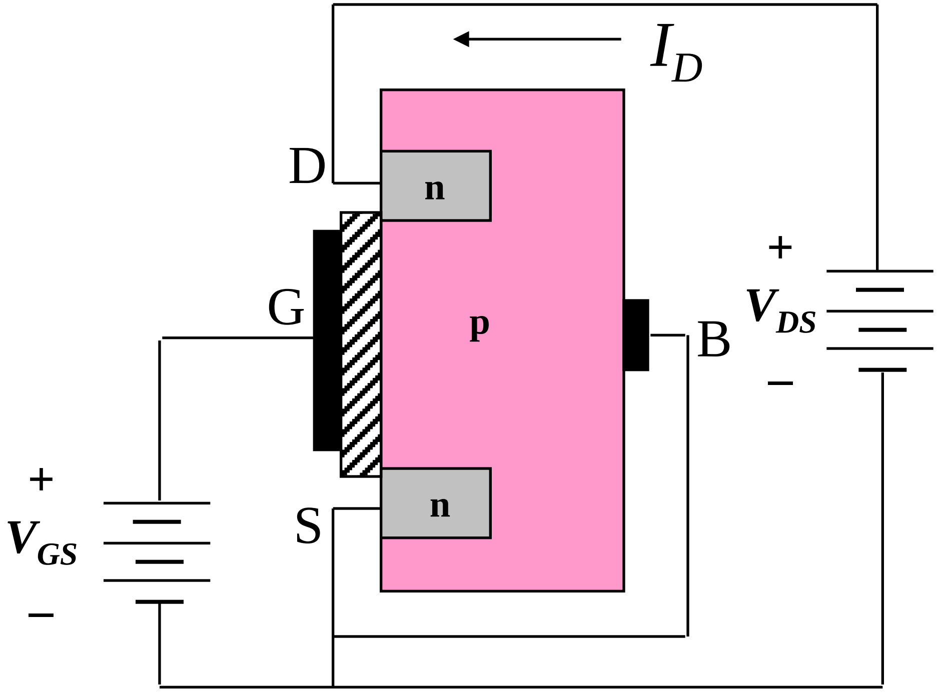
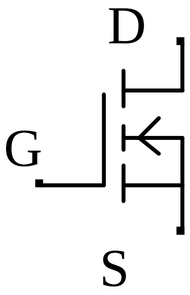
Electronic Circuits Laboratory
EE462G
Lab #5

Biasing MOSFET devices

n-Channel MOSFET

A Metal-Oxide-Semiconductor field-effect transistor (MOSFET) is presented for charge flowing in an n-channel:





B – Body or Substrate

D – Drain

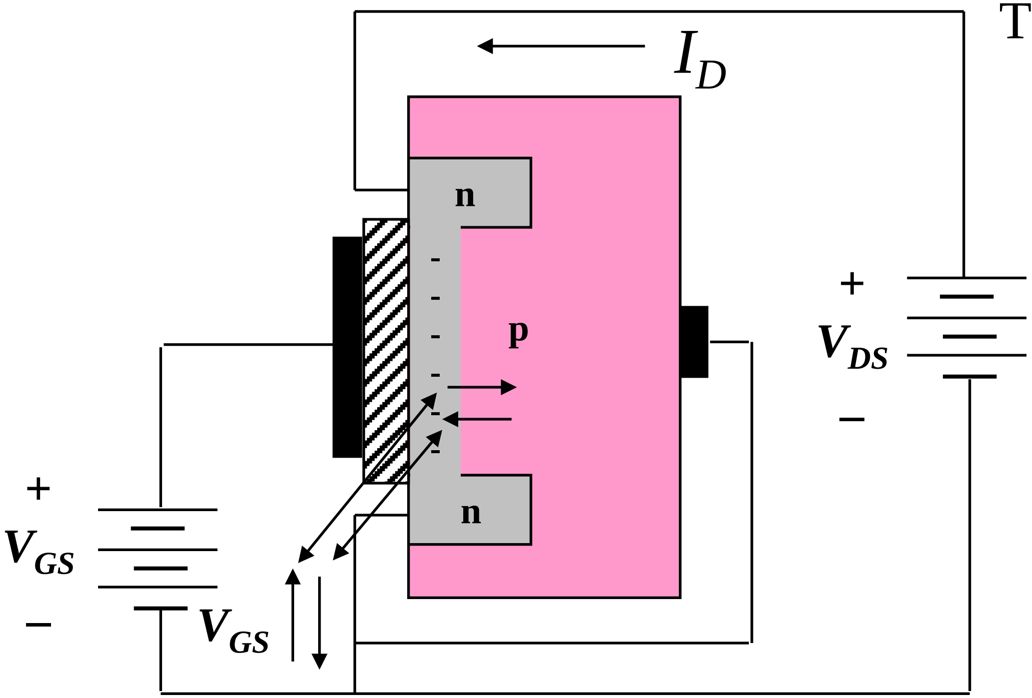
G – Gate

S – Source

For many applications the body is connected to the source and thus most FETs are packaged that way.

FET Operation

The current flow between the drain and the source can be controlled by applying a positive gate voltage:



Three Regions of Operation:

Cutoff region

$(V_{GS} \leq V_{tr})$

Triode region

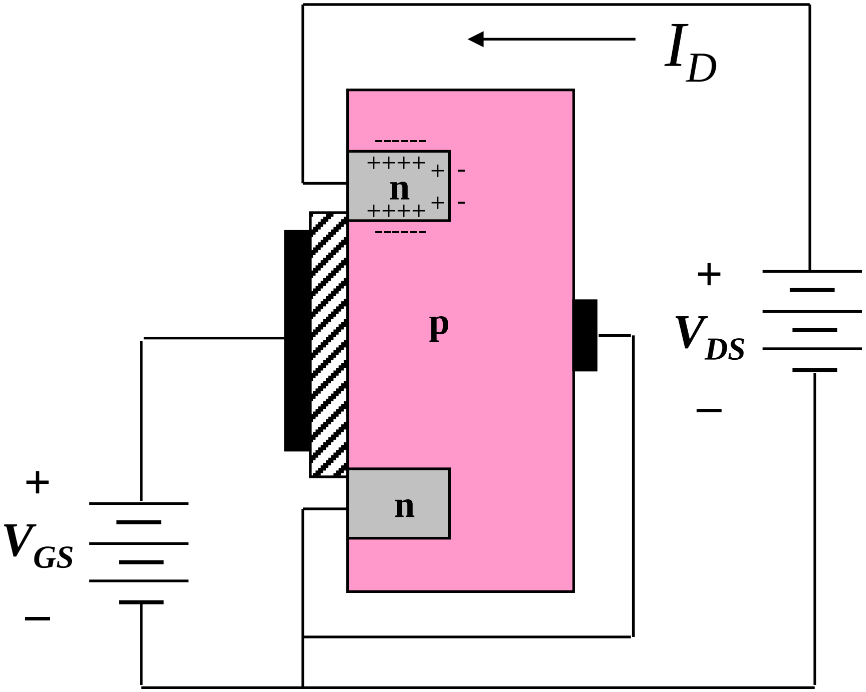
$(V_{GS} \geq V_{tr}\ ,\ V_{DS} \leq V_{GS} - V_{tr}\ )$

Const Current (Saturation) region

$(V_{GS} \geq V_{tr}\ ,\ V_{DS} \geq V_{GS} - V_{tr}\ )$

Cutoff Region

In this region ($V_{GS} \leq V_{tr}$) the gate voltage is less than the threshold voltage and virtually no current flows through the reversed biased PN interface between the drain and body.

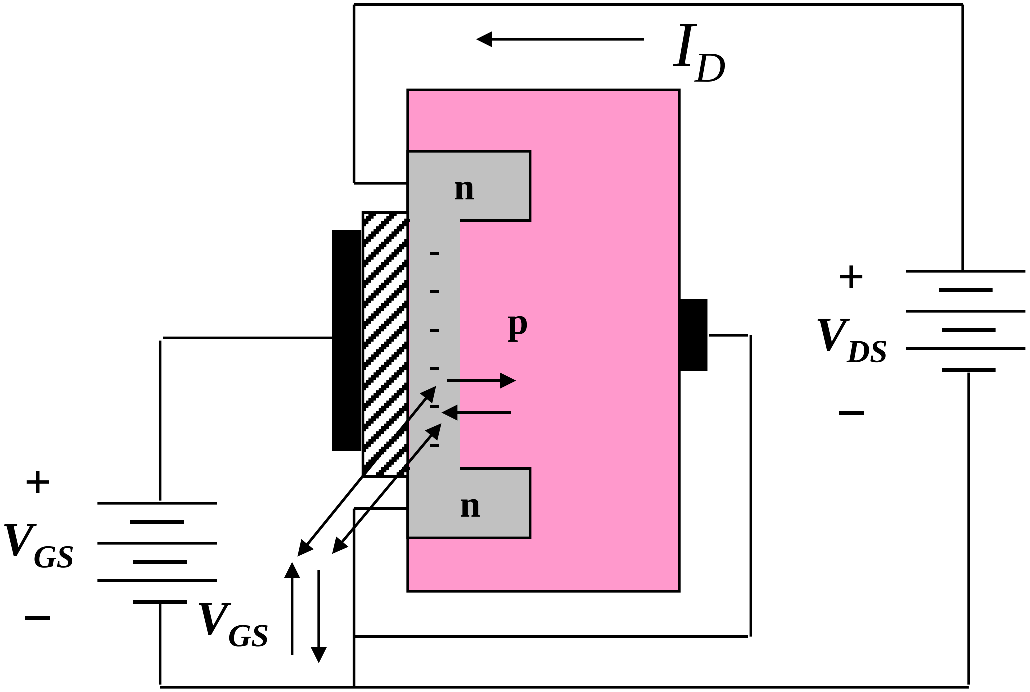


Typical values for V_{tr} (or V_{to}) range from 1 to several volts.

Cutoff region: $I_D=0$

Triode Region

In this region ($V_{GS} > V_{tr}$ and $V_{DS} \leq V_{GS} - V_{tr}$) the gate voltage exceeds the threshold voltage and pulls negative charges toward the gate. This results is an *n*-Channel whose width controls the current flow I_D between the drain and source.



Triode Region:

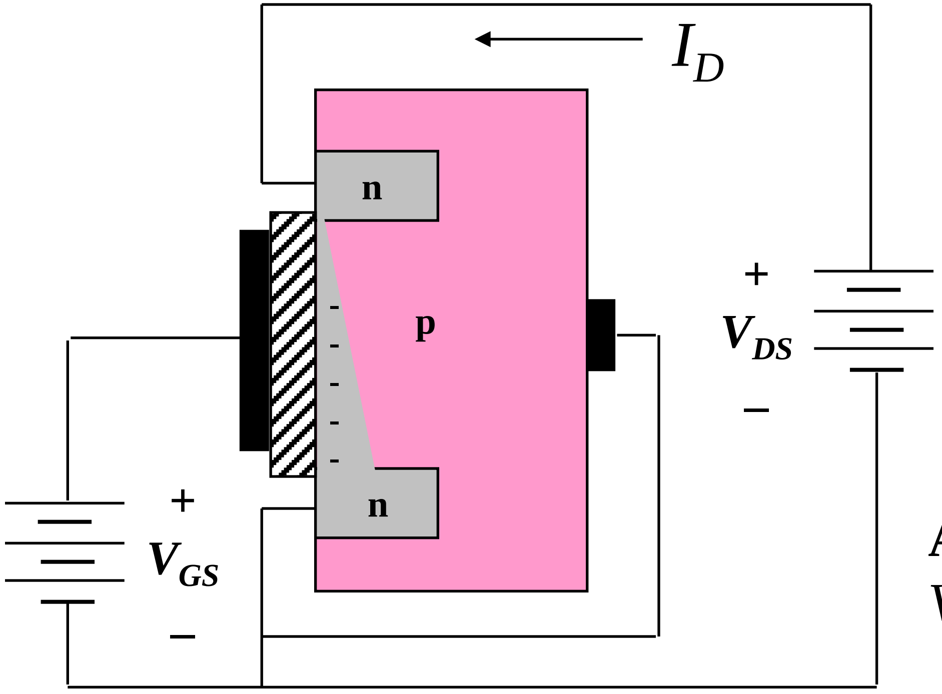
$(V_{GS} > V_{tr}, V_{DS} \leq V_{GS} - V_{tr})$

$$I_D = \left(\frac{W}{L}\right)\frac{K_P}{2}\left[2(V_{GS} - V_{tr})V_{DS} - V_{DS}^2\right]$$

where: $K_P = \mu_n C_{ox}$ product of surface mobility of channel electrons μ_n and gate capacitance per unit area C_{ox} in units of amps per volts squared,
W is the channel width, and
L is channel length.

Constant Current Region

In this region ($V_{GS} > V_{tr}$ and $V_{GS} - V_{tr} \leq V_{DS}$) the drain-source voltage exceeds the excess gate voltage and pulls negative charges toward the drain and reduces the channel area at the drain. This limits the current making it more insensitive/independent to changes in V_{DS}.



Constant current region (CCR):

$V_{GS} - V_{tr} \leq V_{DS}$

$$I_D = \left(\frac{W}{L}\right)\frac{KP}{2}(V_{GS} - V_{tr})^2$$

The material parameters can be combined into one constant:

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

At the point of CCR, for a given V_{GS}, the following relation holds:

$$I_D = KV_{DS}^2$$

NMOS Transfer Characteristics

The relations between I_D and V_{DS} for the operational regions of the NMOS transistor can be used to generate its transfer characteristic. These can be conveniently coded in a Matlab function

```
function ids = nmos(vds,vgs,KP,W,L,vto)

%  Function generates the drain-source current values "ids" for
%  and NMOS Transistor as a function of the drain-source voltage "vds".
%    ids = nmos(vds ,vgs,KP,W,L,vto)
%    where "vds" is a vector of drain-source values
%                "vgs" is the gate voltage
%                "KP" is the device parameter
%                "W" is the channel width
%                "L" is the channel length
%                "vto" is the threshold voltage
%   and output "ids" is a vector of the same size of "vds"
%   containing the drain-source current values.
```

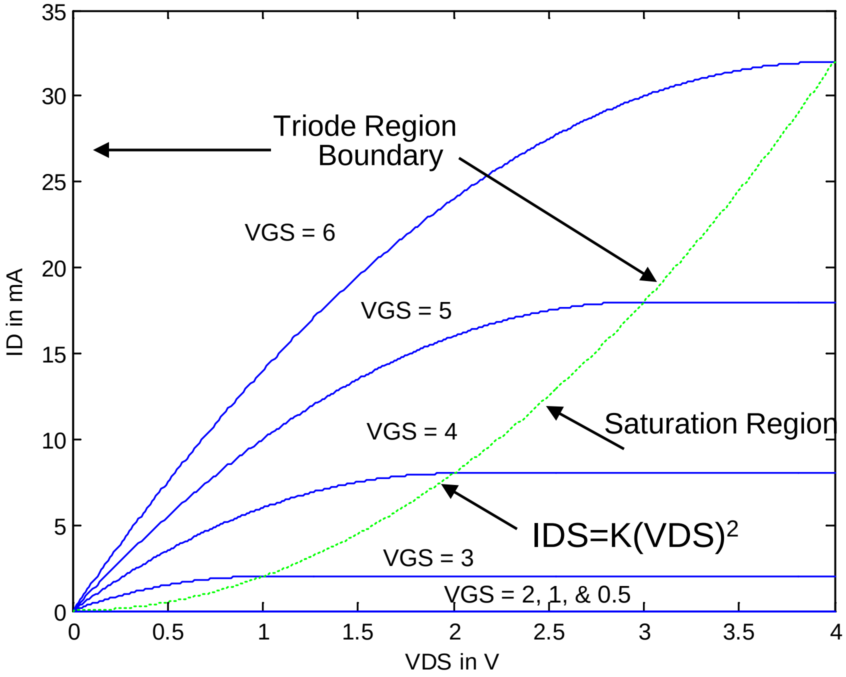
NMOS Transfer Characteristics

```
ids = zeros(size(vds));  %  Initialize output array with all zeros
k = (W/L)*KP/2;   %  Combine devices material parameters
%  For non-cutoff operation:
if vgs >= vto
    %  Find points in vds that are in the triode region
    ktri = find(vds<=(vgs-vto) & vds >= 0);  %  Points less than (gate – threshold voltage)
    %  If points are found in the triode region compute ids with proper formula
    if ~isempty(ktri)
       ids(ktri) = k*(2*(vgs-vto).*vds(ktri)-vds(ktri).^2);
    end
    % Find points in staturation region
    ksat = find(vds>(vgs-vto) & vds >= 0);   % Points greater than the excess voltage
    %  if points are found in the saturation region compute ids with proper formula
    if ~isempty(ksat)
       ids(ksat) = k*((vgs-vto).^2);
    end
 %  If points of vds are outside these ranges then the ids values remain zero
end
```

NMOS Transfer Characteristics

Plot the transfer characteristics of an NMOS transistor where KP = 50 µA/V^2, W= 160 µm, L= 2 µm, V_{tr}= 2V, and for VGS = [.5, 1, 2, 3, 4, 5, 6] volts. Also plot boundary between the saturation and triode regions

```
vgs = [.5, 1, 2, 3, 4, 5, 6];
vds =[0:.01:4];
for kc = 1:length(vgs)
  ids = nmos(vds,vgs(kc),50e-6,160e-6,2e-6,2);
  figure(1); plot(vds, ids*1000)
  hold on
end
ids = (50e-6/2)*(160e-6/2e-6)*vds.^2;
figure(1); plot(vds, ids*1000,'g:')
hold off
xlabel('VDS in V')
ylabel('ID in mA')
```



Triode Region
Boundary

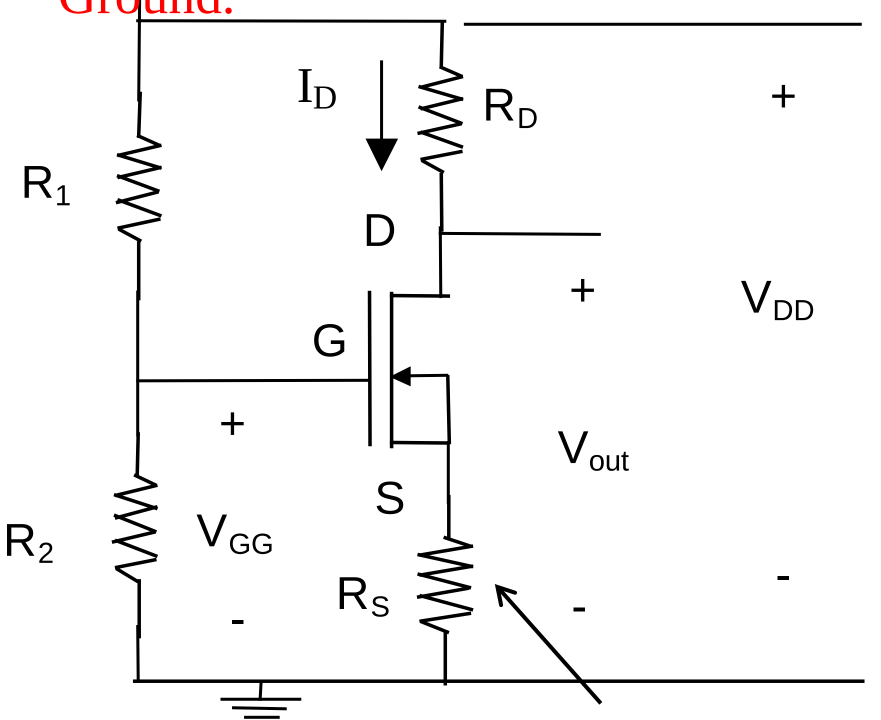
Saturation Region

IDS=K(VDS)^2

Biasing NMOS Voltages

KVL can be applied to the following circuit to determine resistor values so that V_{DS} and I_D are set to a desired quiescent point.

What would happen for V_{out} if there is no Rs? After adding Rs, what will happen for V_{out}? What is called when adding Rs bewteen the Source and Ground.



Feedback bias

KVL

$$V_{DD} - V_{DS} = (R_D + R_S)I_D$$

$$\frac{V_{DD}}{(R_D + R_S)} - \frac{1}{(R_D + R_S)}V_{DS} = I_D$$

The above linear equation (**Load line**) can be plotted with the MOSFET's TC to perform a load line analysis and find the quiescent point V_{DSQ} and I_{DQ}.

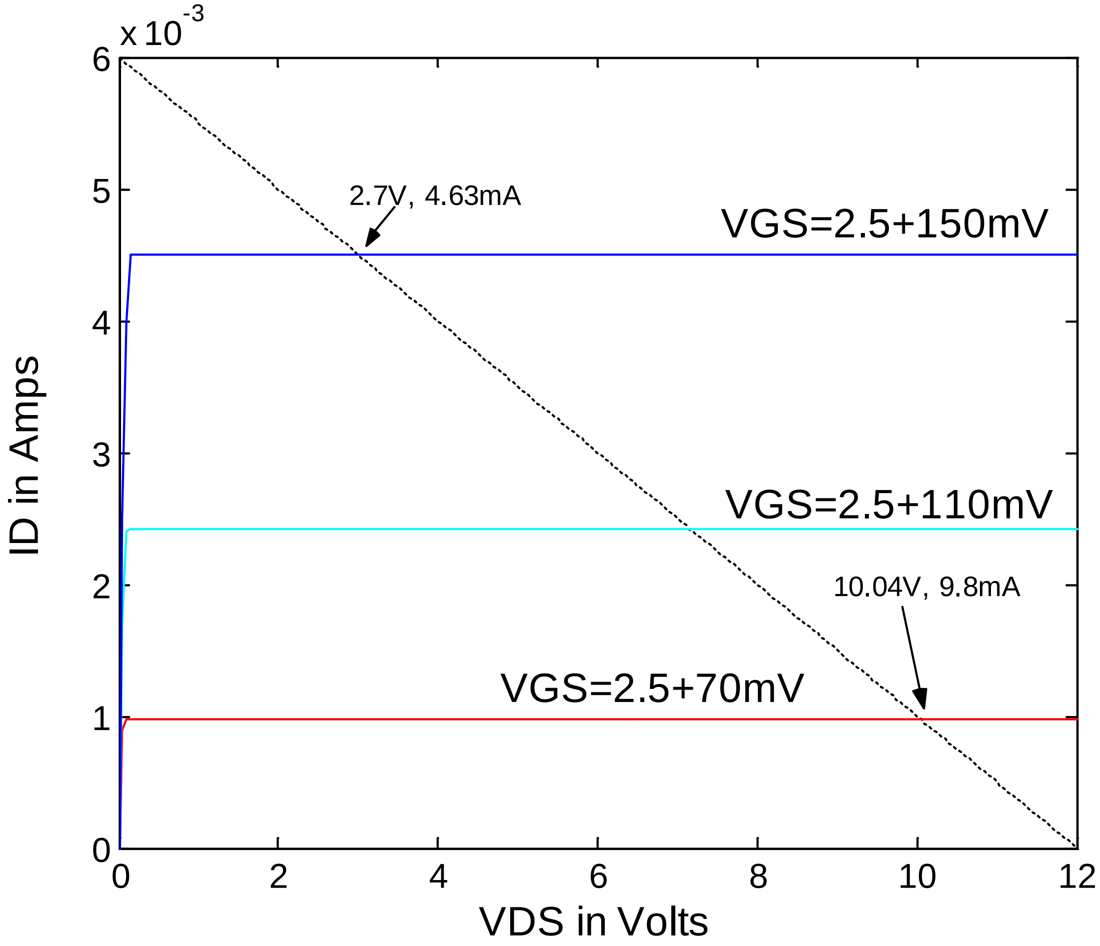
How to do load line analysis analytically if V_{DD}=10 Volts, R_D=5k Ohm and R_S=5k Ohm?

Load Line Analysis Example

Given $R_D = R_S = 1\text{k}\Omega$ and $V_{DD} = 12$ V, superimpose loadline on the TC for various V_{GS} values of MOSFET with K= 0.2 A/V^2 and $V_{tr} = 2.5$ V

```
>> %  Set Parameters
>> K=.2; vto = 2.1;
>> W=1; L=1; KP=2*K;
>> VDD=12;  RS=1e3; RD=1e3;
>> vds = [0:.05:VDD];  %  Create X-Axis
>> idsll = -vds/(RD+RS) + VDD/(RD+RS);  %  Generate Load Line
>> plot(vds, idsll, 'k:')
>> hold on   %  hold plot to superimpose other plots
>> ids50 = nmos(vds,vto+70e-3,KP,W,L,vto);  % TC for 70mV above threshold
>> plot(vds,ids50,'r')
>> ids50 = nmos(vds,vto+110e-3,KP,W,L,vto);  % TC for 110mV above threshold
>> plot(vds,ids50,'c')
>> ids50 = nmos(vds,vto+150e-3,KP,W,L,vto);  % TC for 150mV above threshold
>> plot(vds,ids50,'b')
```

Load Line Analysis Example



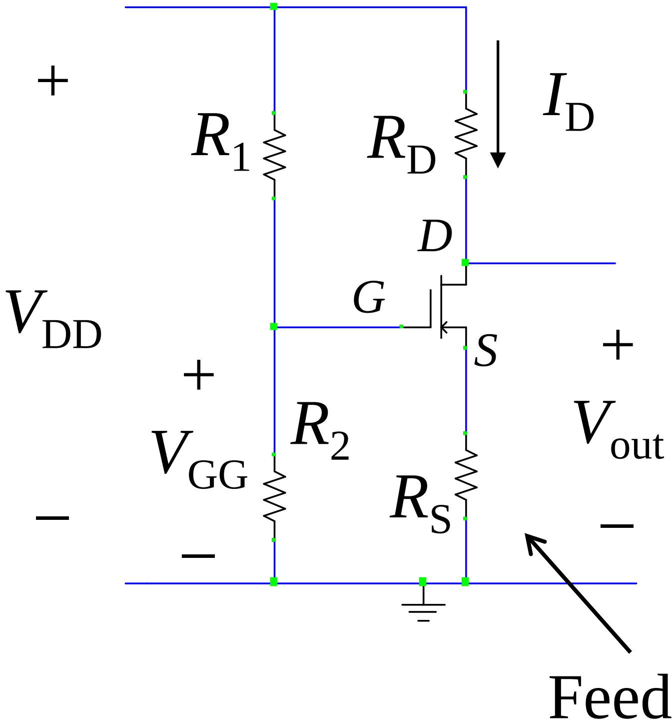
If changes about V_{GSQ} are considered the input signal, and changes in V_{DSQ} are considered the output signal, then the Gain is:

$$G_V = \frac{\Delta V_{DS}}{\Delta V_{GS}} = \frac{(2.7-10.04)\text{V}}{(150-70)\text{mV}}$$

$$G_V = -91.75$$

Biasing NMOS Voltages

KVL can be applied to relate the gate voltage to the drain-source currents:



Since virtually no current flows into the gate, V_{GG} can be set by properly choosing R_1 and R_2. KVL around the V_{GG} loop yields another important equation:

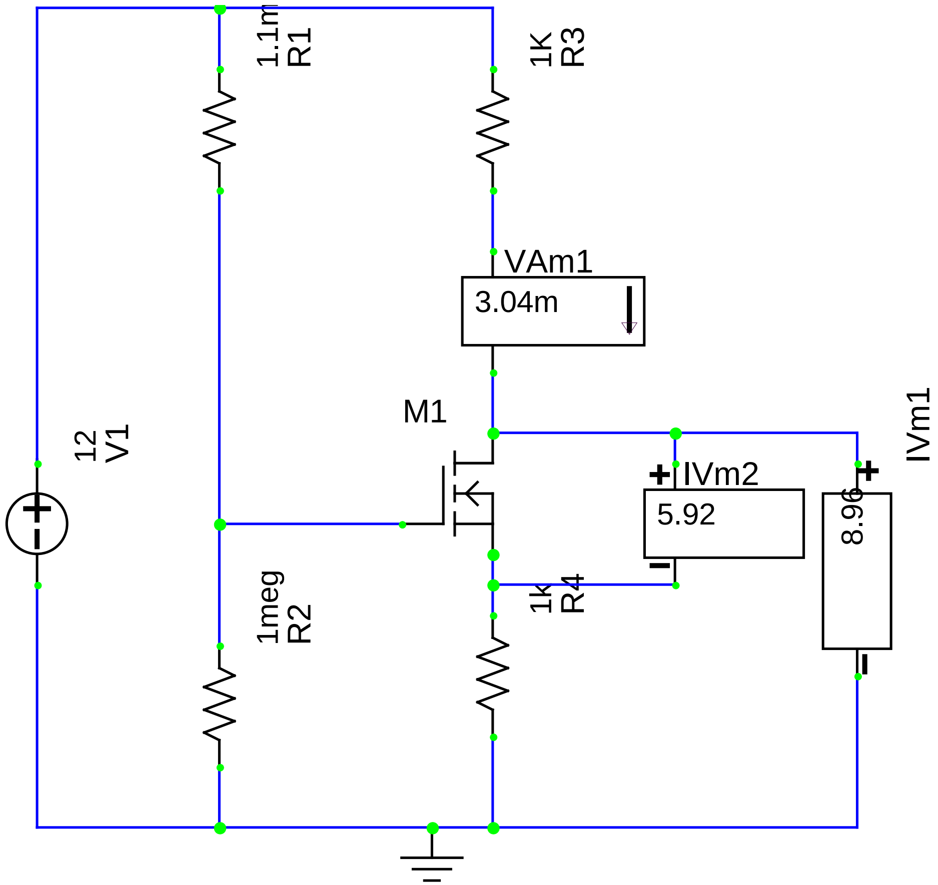
$$V_{GG} - V_{GS} = R_S I_D$$

When biasing in the saturation region V_{GS} can also be related to the drain current by:

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

SPICE Analysis

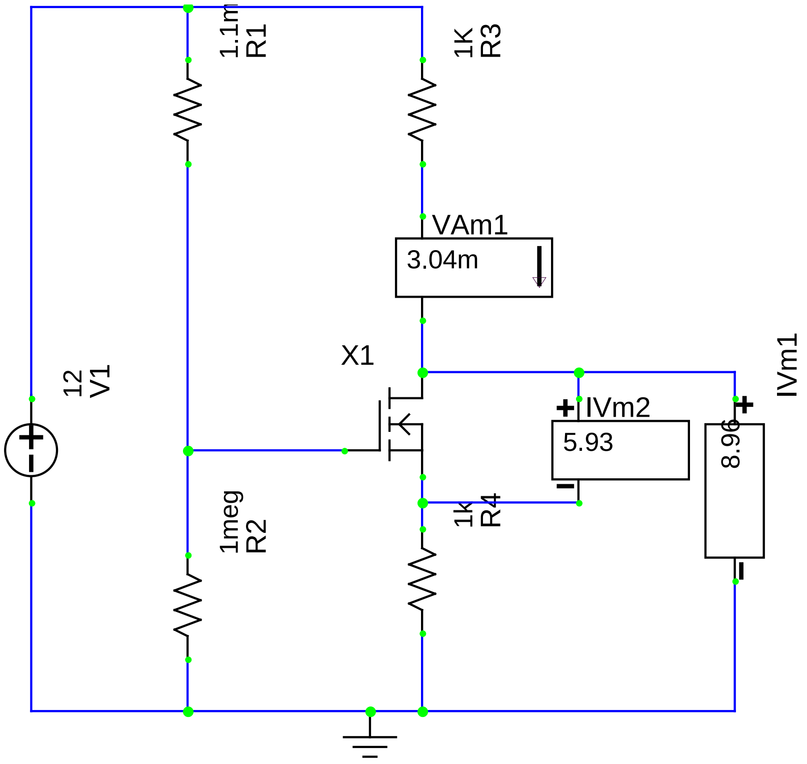
Previous load line analysis may suggest setting the DC operating point at I_{DSQ}=3mA and V_{DSQ}=6 (center of load line). This requires that V_{GSQ} be set around 2.62 V, which results in the following circuit:



An operating point analysis in SPICE can be performed to verify the quiescent points throughout the circuit. The new aspect of the SPICE feature for this work is setting the MOSFET parameters. In this case the a level 1 nmos transistor can be selected from the device menu and the *Kp* and *vto* values set through the edit simulation parameters option (then click on shared properties tab).

SPICE Analysis

The MOSFET can be selected directly as the ZVN3306, which can be accessed through the “Browse for Parts” option under the “Devices” menu. Then on the edit simulation model, the parameters of V_{to} = 2.5 and KP=.2 can be set in the text-like file that appears (SPICE model in netlist syntax).

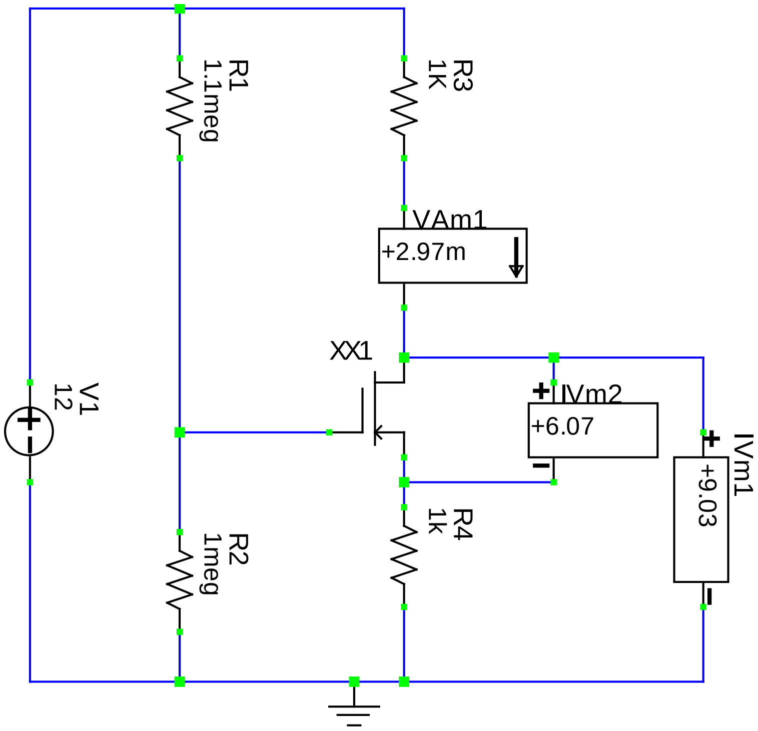


Run operating point analysis and observe quiescent values in the meters.

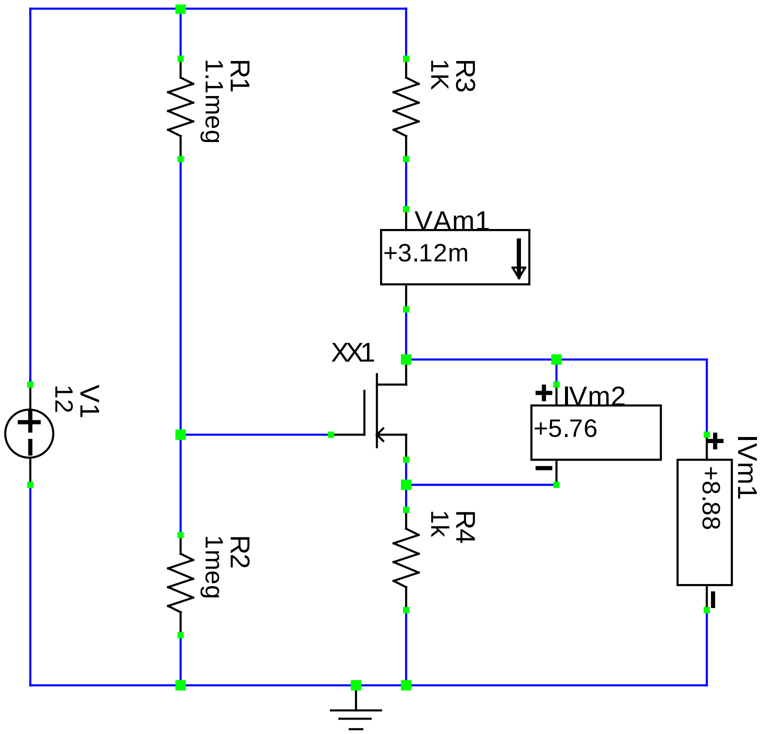
SPICE Analysis

Change to KP=.1 and then to .8 to observed relative changes in the operating point.

KP=.1



KP=.8



Iteration in Matlab

Matlab has 2 main instructions for setting up loops - the WHILE and FOR loops. Type *help while* or *help for* for information on how to use either of these.

Iteration in Matlab

To find the intersection between 2 curves, represented by sampled vectors, use the *min* command to find the CLOSEST 2 points:

```
>> [minval, minindex] = min(abs(tc-loadline));
```

The *min* command finds the minimum value in the vector. If a second output is requested as in the above expression, the index of where this minimum occurs is assigned to the second input. In the above expression *minval* is the minimum value of the point by point absolute difference between vectors *tc* and *loadline, and minindex* is the vector index at which this minimum occurs. Type *help min* for more information.