Electronic Circuits Laboratory EE462G Lab #2

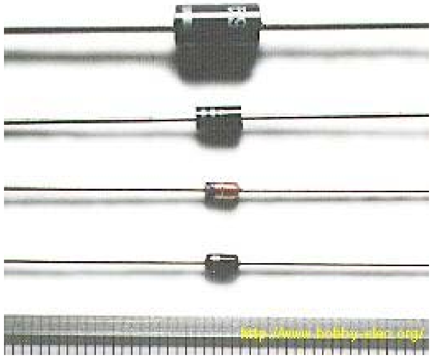
Characterizing Nonlinear Elements, Semiconductor Parameter Analyzer, Transfer Characteristics,

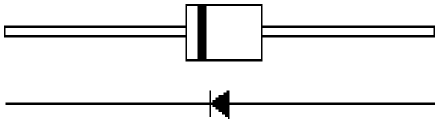
Curve Fit Programs

Original slides prepared by Kevin D. Donohue (Spring 2007)
Modified by Zhi David Chen (Fall 2018)

Nonlinear Elements

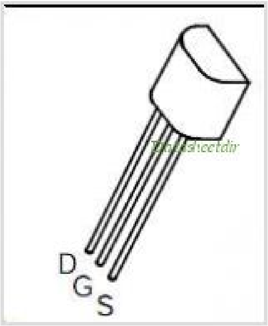
Diodes





- *pn* junction
- Zener

Transistors

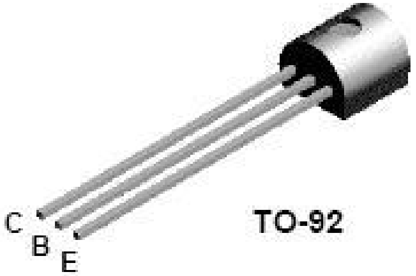


- Field Effect Transistors – (FETs)

  ZVN3306A -NMOSFET
  ZVP3306A -PMOSFET

- Bipolar Junction Transistors – (BJTs)

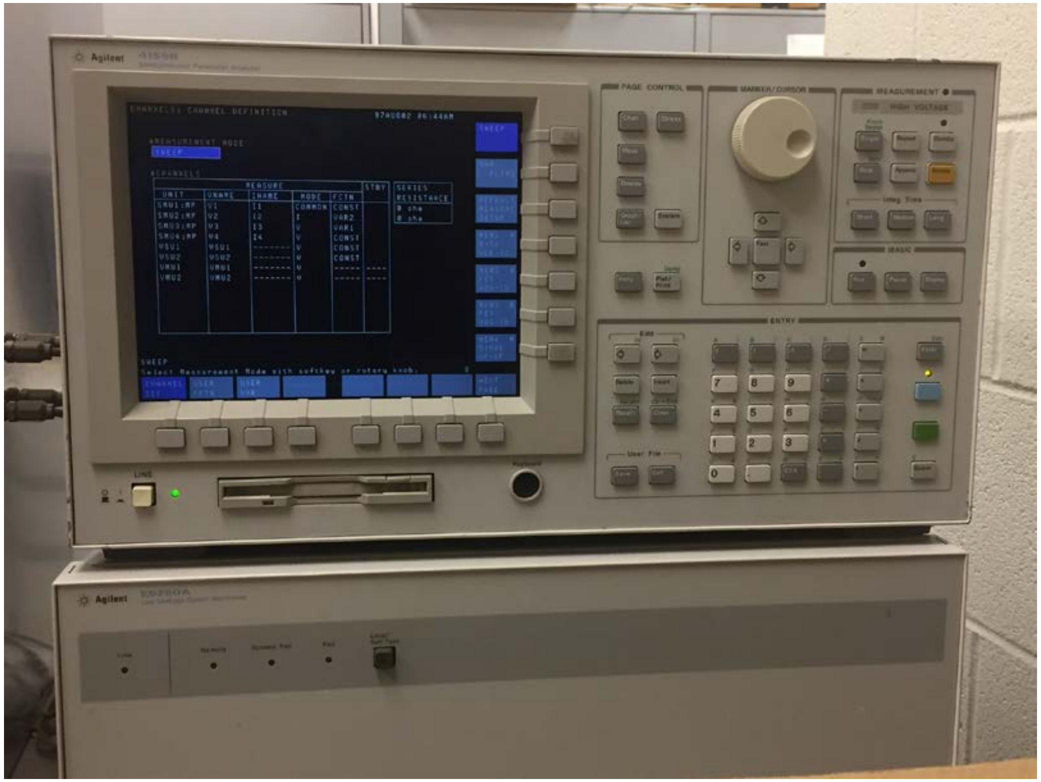
  PN2222 –NPN BJT



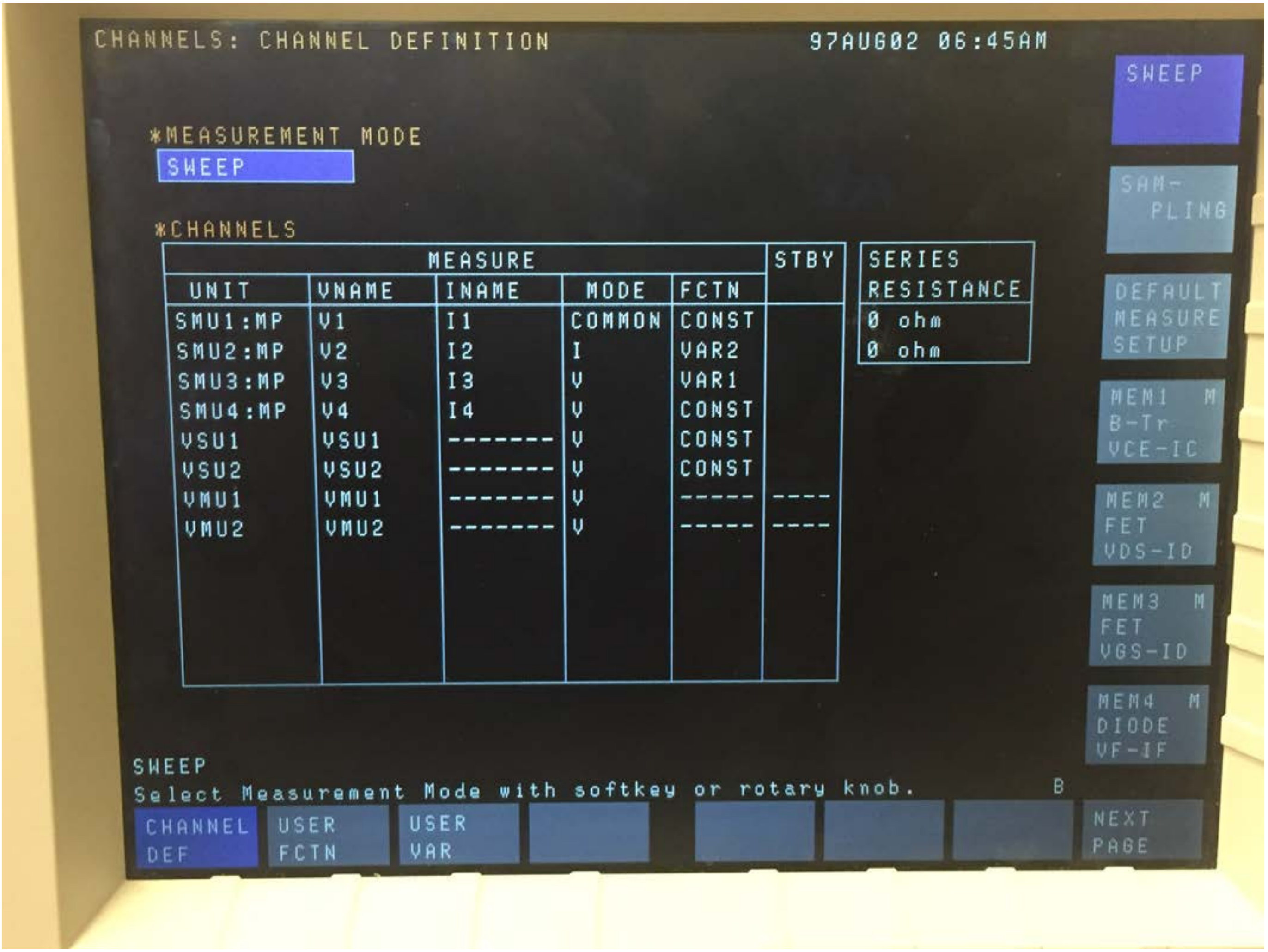
Instrumentation

This lab requires:

- Agilent 4155B Semiconductor Parameter Analyzer at 349 ASTeCC
- Because the floppy drive does not work anymore. For the data sheet you must take photos of the curves. In the lab report you can include the photos for your curves.

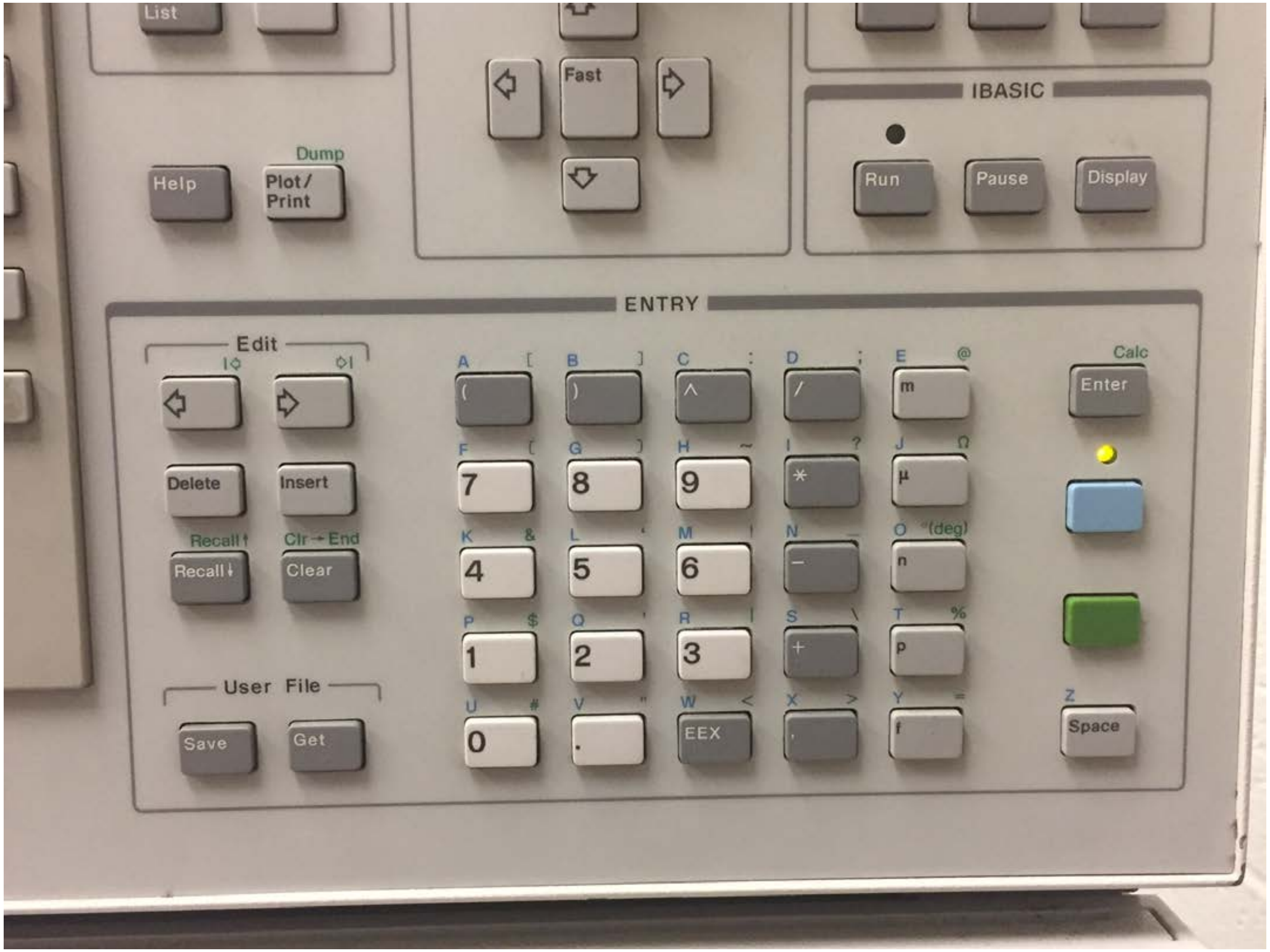


Screen Display



Control and Measurement Panels

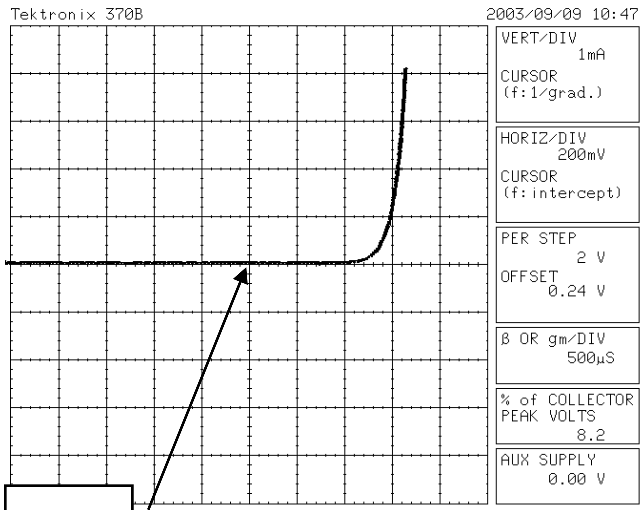
Entry Panel



List
Fast
IBASIC
Run
Pause
Display
Help
Dump
Plot / Print
ENTRY
Edit
Delete
Insert
Recall↑
Clr→End
Recall↓
Clear
User File
Save
Get
Calc
Enter
EEX
Space

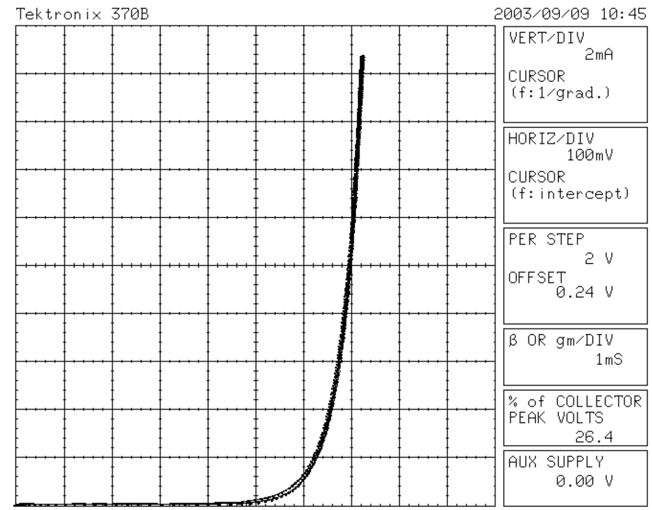
Instrumentation

Example displays for diode with different collector settings



0,0

AC collector sweep

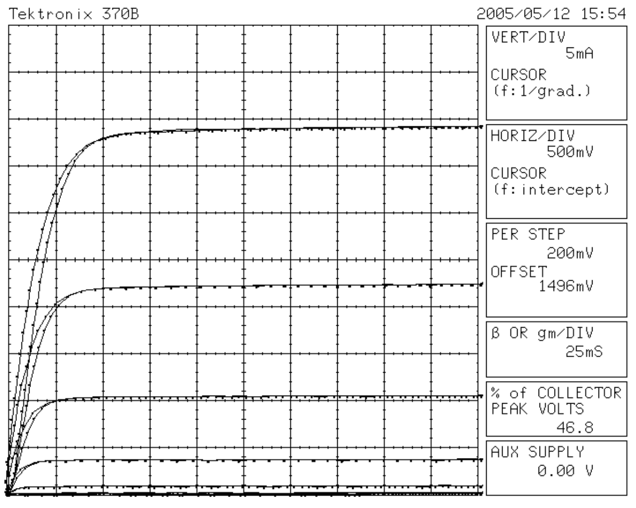


Positive DC collector sweep

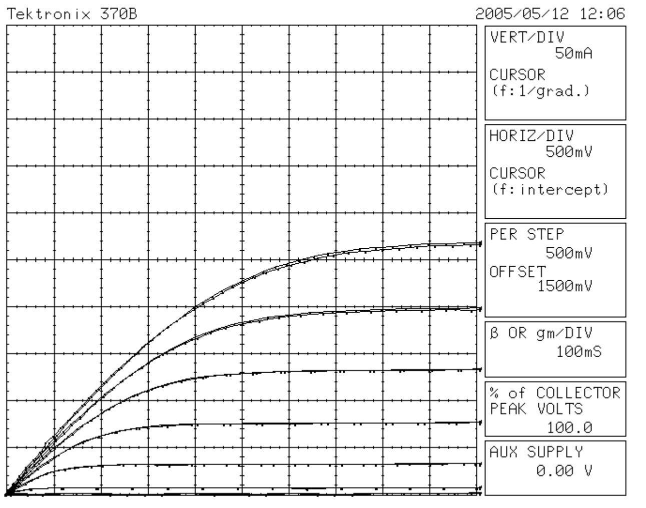
0,0

Instrumentation

Example displays for FET drain characteristics with different gate-source voltage steps and horizontal and vertical scales.



Offset 1.492 V
Step .2 V
Step Number 7



Offset 1.5 V
Step .5 V
Step Number 7

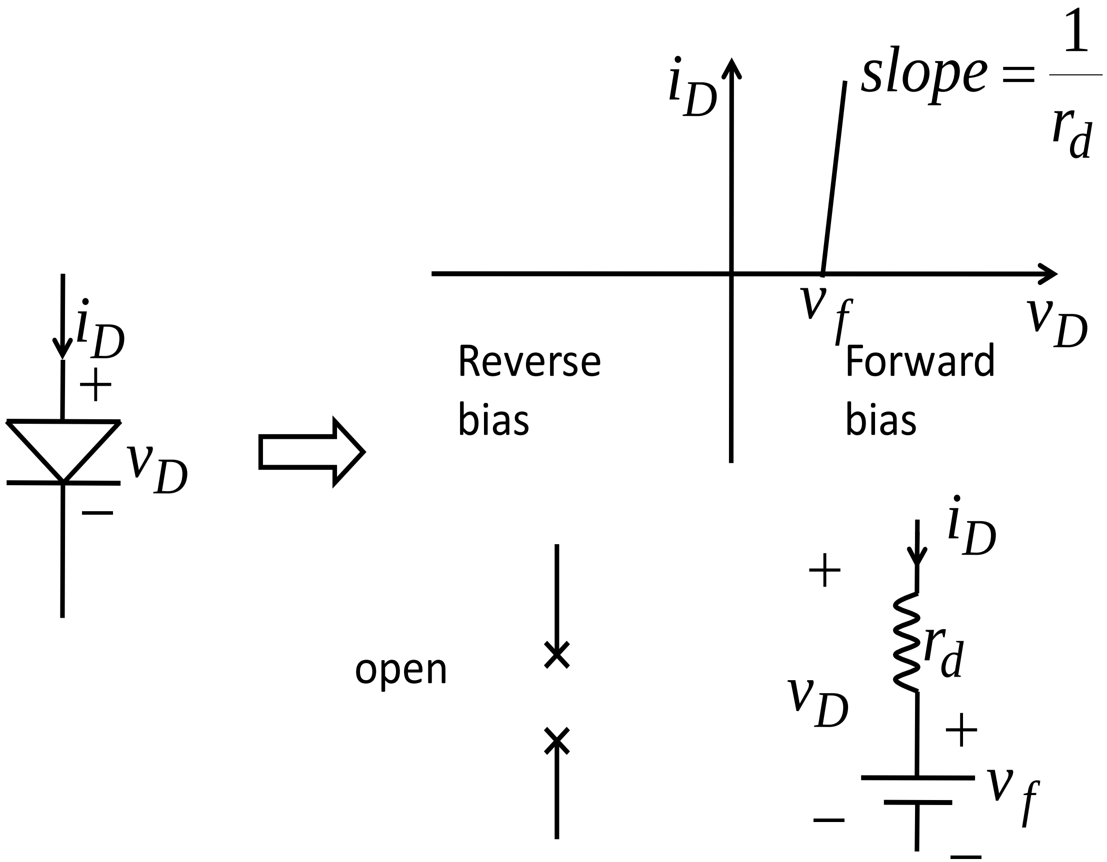
Transfer Characteristics

Transfer Characteristics

- A relation describing the amplitude input-output relationship of a device.

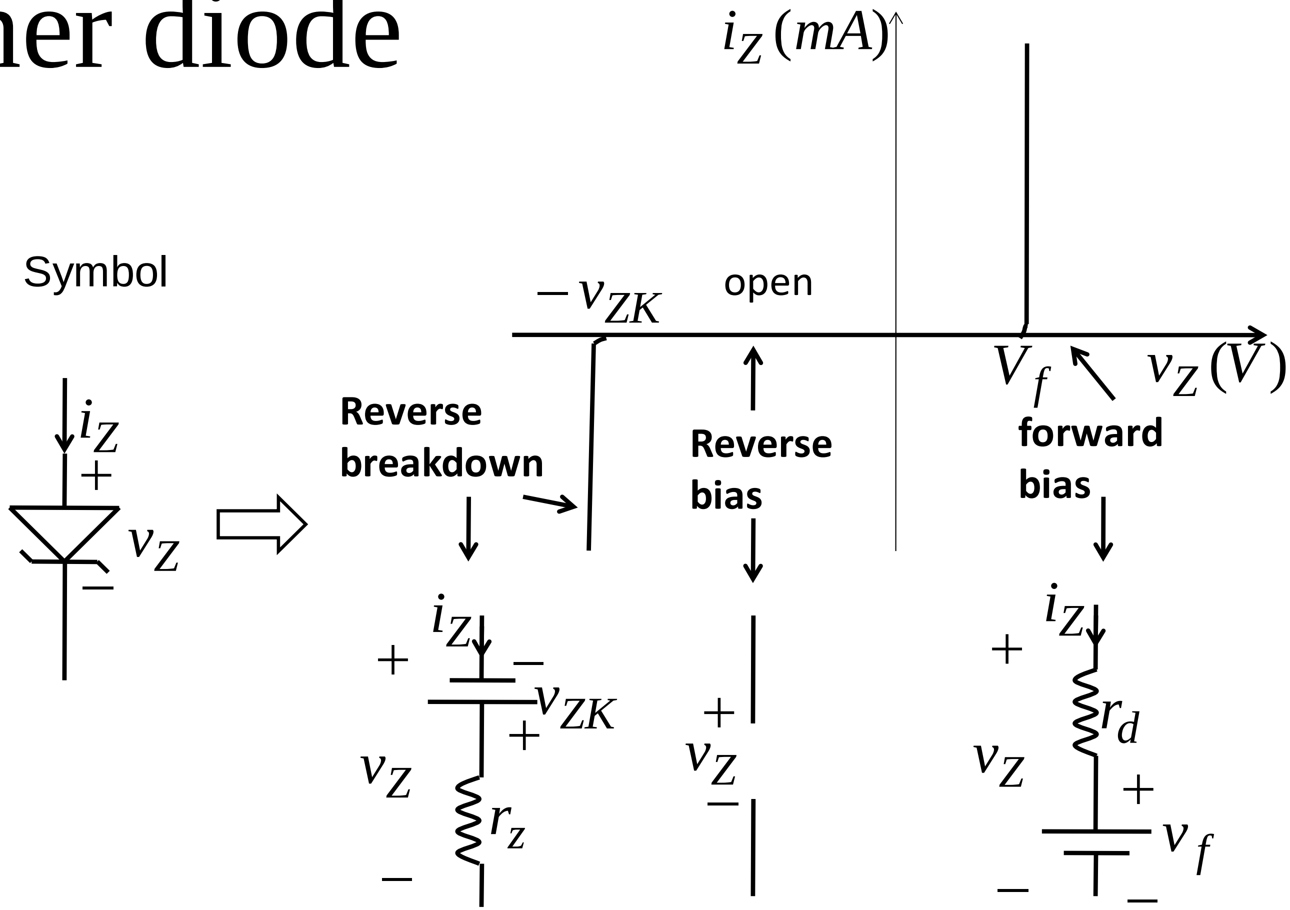
- A sufficient characterization in most engineering problems for instantaneous systems (present output does not depend on future or previous values). These systems are sometimes referred to as memoryless, and are typical of systems with no energy storage elements.

PN junction diode



The band on the component usually denotes the cathode terminal

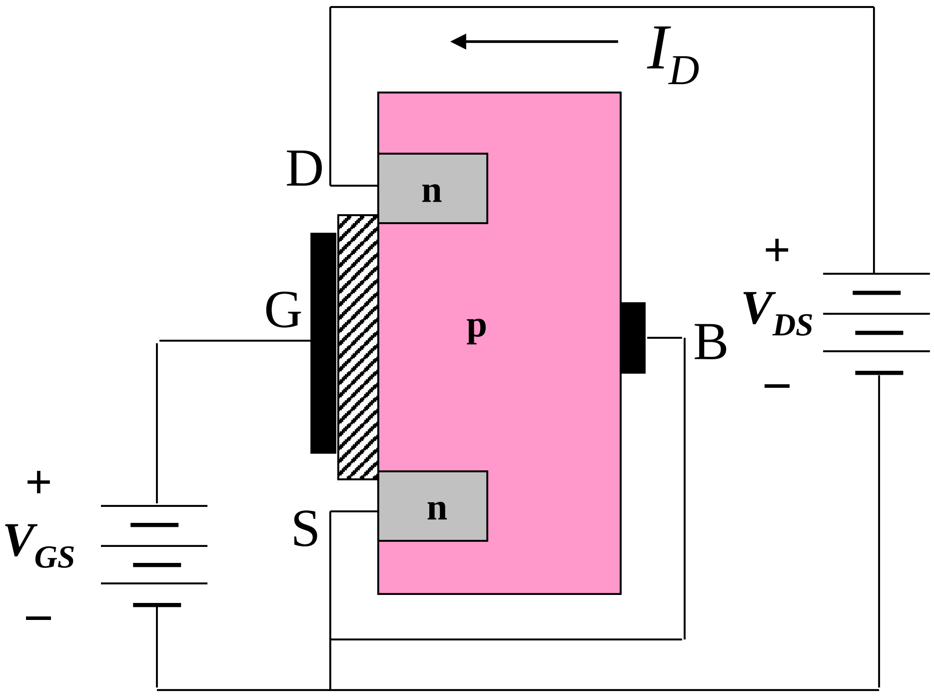
Zener diode

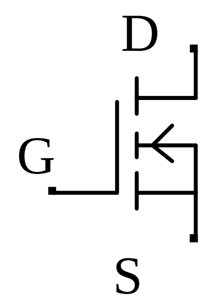


The band on the component usually denotes the cathode terminal

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

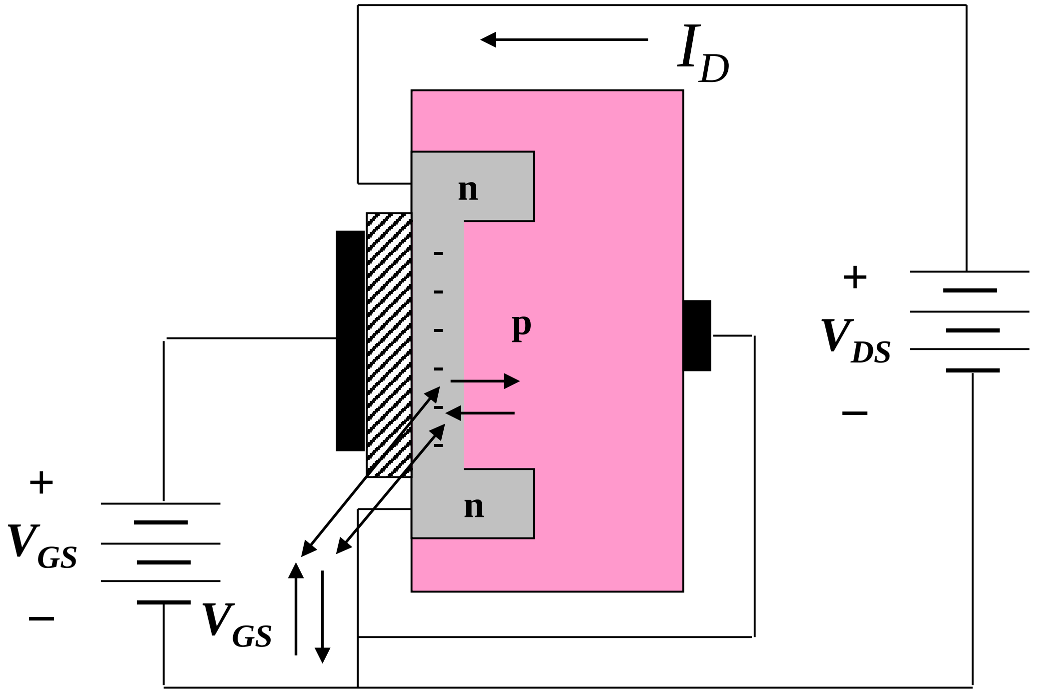
G – Gate

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

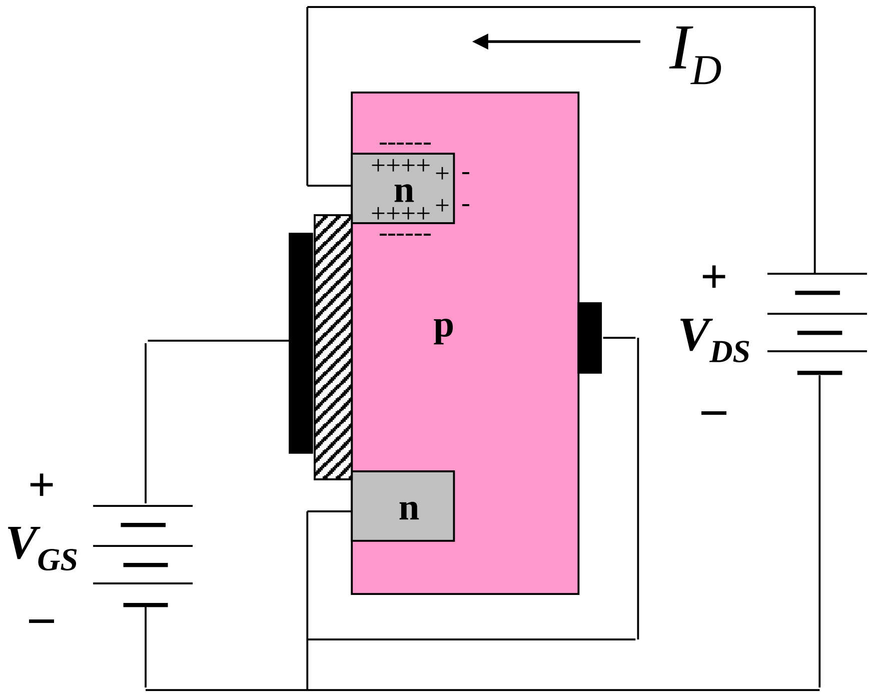
$(V_{GS} \geq V_{tr}\,,\ V_{DS} \leq V_{GS} - V_{tr})$

Constant current region (CCR) (saturation)

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

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.

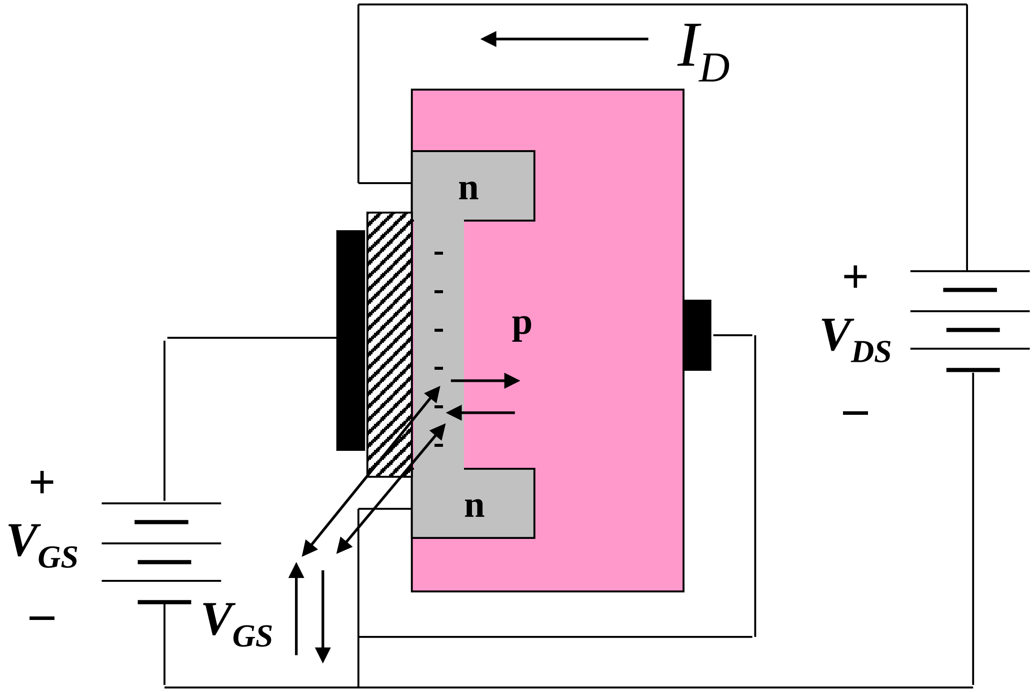


Typical values for V_{tr} (or V_{to}) range from 1 to several volts.

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:

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

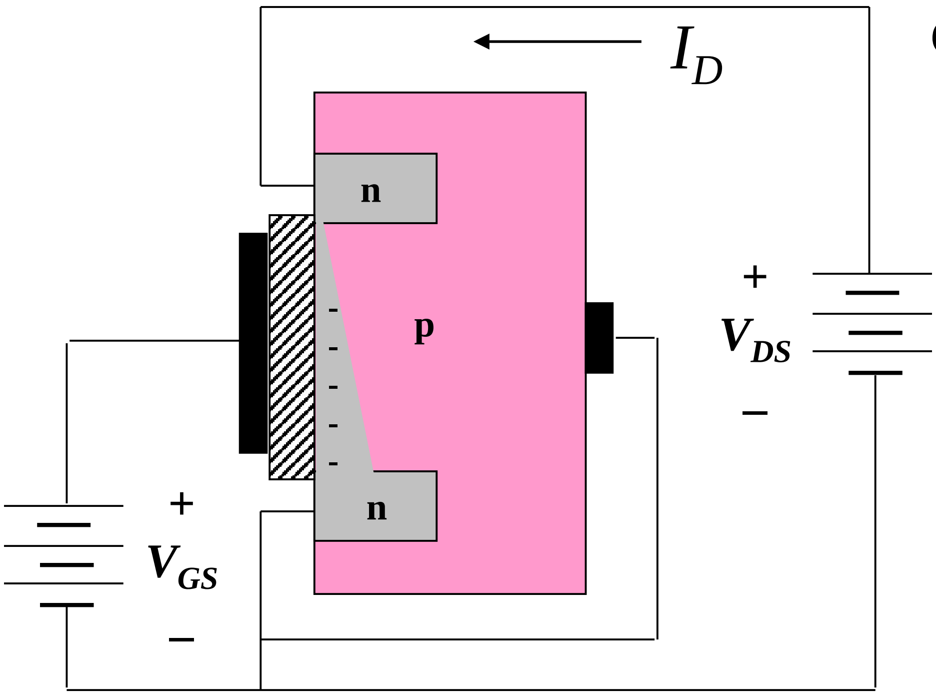
$$I_D = \left(\frac{W}{L}\right)\frac{K_p}{2}\left[2(V_{GS} - V_{tr})V_{DS} - V_{DS}^2\right]$$

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 (CCR)

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}.



CCR: $V_{GS} > V_{tr}$, V_{GS} - $V_{tr} \leq V_{DS}$

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

The material parameters can be combined into one constant:

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

At the point of beginning of CCR (Saturation), 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)

%  This 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.
```

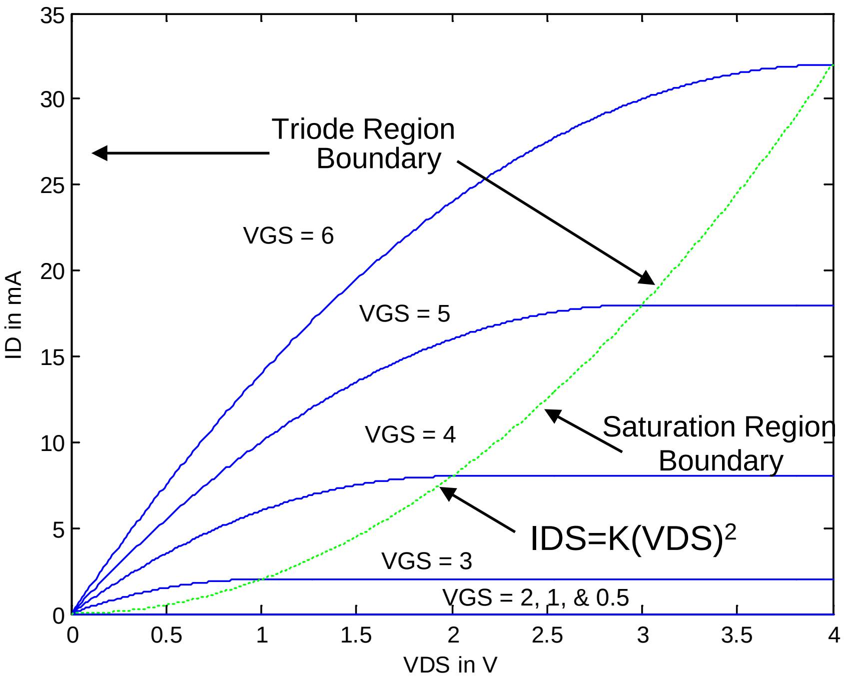
NMOS Transfer Characteristics

```
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 saturation region
    ksat = find(vds>(vgs-vto) & vds >= 0);   % Points greater than the excess voltage
    %  if points are found in the saturation regions 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

```
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')
```



Find K from Curve Trace

Recall at the start of CCR the following holds: $I_D = KV_{DS}^2$

From graph: $I_D = KV_{DS}^2$

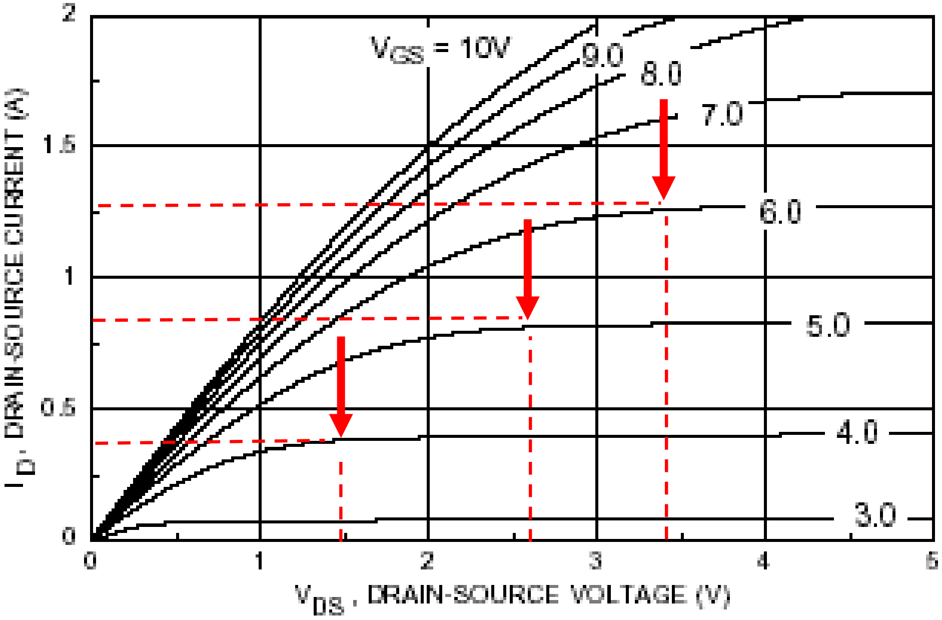
$0.4 = K1.5^2$

$0.7 = K2.6^2$

$1.5 = K3.4^2$

One way to estimate K is to compute it for each equation and take the average. In this case it becomes K=.137 A/V^2, which for W=L=1, implies:

K_p= K*2 = 0.274 A/V^2



Find K from Curve Trace (LS)

From graph:

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

$$0.4 = K1.5^2$$

$$0.7 = K2.6^2$$

$$1.5 = K3.4^2$$

A better approach uses a least-squares solution that gives the error in each equation equal weight. Find K to minimize mean square error for the measured data:

$$\left\langle E^2 \right\rangle = \frac{1}{M}\sum_{i=1}^{M}\left(I_{Di} - KV_{DSi}^2\right)^2$$

Take derivative with respect to K and set equal to 0 to obtain:

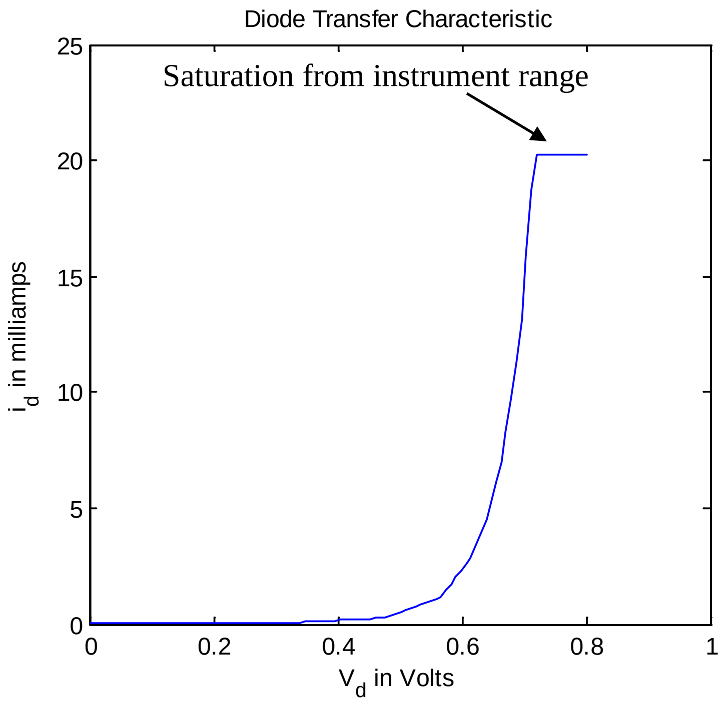
$$K_{LS} = \frac{\sum_{i=1}^{M}\left(V_{DSi}^2 I_{Di}\right)}{\sum_{i=1}^{M}\left(V_{DSi}^2 V_{DSi}^2\right)} \quad K_{LS} = \frac{\left(.4\times 1.5^2 + .7\times 2.6^2 + 1.5\times 3.4^2\right)}{\left(1.5^2\times 1.5^2 + 2.6^2\times 2.6^2 + 3.4^2\times 3.4^2\right)} = .1246 \quad \frac{\mathrm{A}}{\mathrm{V}^2}$$

which for W=L=1, implies: $K_p = K_{LS}*2 = 0.2492$ A/V^2

Plot Diode TC

Data was collected on the Tektronix's Curve Tracer 370B for a Diode and saved as a CSV file. Open file, process data and plot TC

```
% Open CSV curve from Curve Tracer
   Output File
fname = ['E121600E.CSV'];
c = getcurves(fname);
% Convert Cell Array to vector
[id, vd] = interpcurves(c);
% Plot it for observation
figure(1)
plot(vd,id*1000)
title('Diode Transfer Characteristic')
xlabel('V_d in Volts')
ylabel('i_d in milliamps')
```



Curve fit Parameter Estimation

Given the TC curve for the diode in terms of its I-V values sampled and stored:

- a straight-forward iterative program can be written to vary parameters in the diode equation and compare its fit (error) to the measured data
- The parameters that result in the minimum error can then be used to described the device.
- The diode can be described with the Shockley equation:

$$i_D = I_s\left[\exp\left(\frac{v_D}{nV_T}\right) - 1\right]$$

- where v_D and i_D is the voltage drop over and current through the diode, respectively, I_s is the saturation current (typically on the order of 10^{-16}), n is the emission coefficient taking on values between 1 and 2, and V_T is the thermal voltage (about equal to 0.026 at 300K).

Curve fit Parameter Estimation

Matlab Example: Consider a parametric variation of the Shockley equation where the product of n and V_T is taken as one parameter:

$\alpha = n \times V_T$

$$i_D = I_s\left[\exp\left(\frac{v_D}{\alpha}\right) - 1\right]$$

- Assume V_T = .026, write a Matlab script to iterate with a range of values for n and I_s while comparing the squared error between the measure i_D values and those predicted with the equation. Find the n and I_s that result in the minimum error.
- The I_s range can be large so you may want to create range of values uniformly spaced on a log scale. This can be done with the Matlab command logspace:
  >> isvec = logspace(-18, -13,150);
  Type help logspace in Matlab for information on how this function works.

```
%  Select parameters ranges and increments
vt = .026;  %  Thermal voltage
nt = [.5:.1:3];   %  Trial values for emission coefficient n
is = logspace(-12,-5,35);  %  Trial values of Is

%  Extract a curve from Curve Tracer Output
c = getcurves(['E121600E.CSV']);
%  Interpolate curve to a uniform set of points for x and y axes of TC
[m, x] = interpcurves(c);

%  Trim data and Only use values less than .01 volt, if you look at a plot of the data
%  you will see it saturates close to 20 mA
idlimit = .0175;
dd = find(m < idlimit);  %  Find all points not affected by saturation
cv = m(dd);              %   Trim data vector to just these points
vds = x(dd);             %   Trim corresponding voltage axis
```

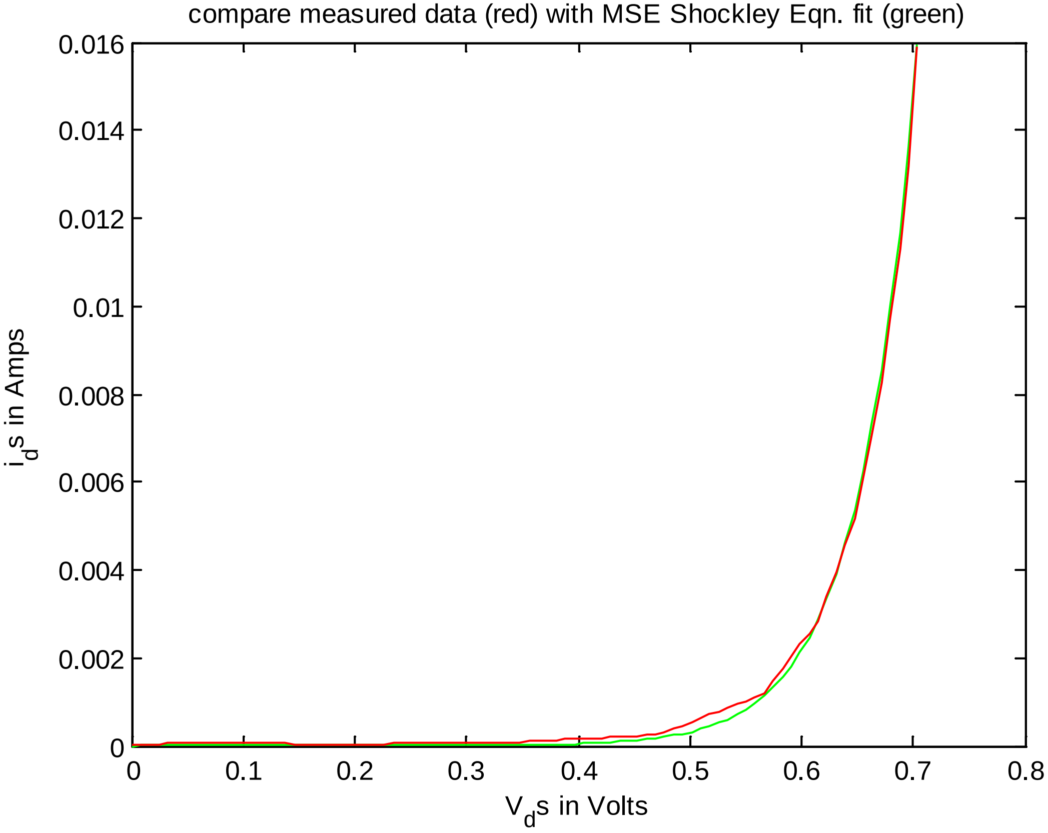
```
%  Loop to compute squared error for every iteration of test value
%  parameter
for n=1:length(nt)   %  Emission coefficient "n" loop
    for k=1:length(is)   %  Saturation current value loop
        ids = diodetc(vds,is(k),nt(n),vt);  %  generate Shockley equation values at measured
                                           %  x-axis values (vds) and trial parameters (you need to
                                            %   create this function as part of your prelab assignment).
        err(n,k) = mean(abs(ids - cv).^2);   %  Compute MSE for this parameter combination
         plot(vds,ids,'g',vds,cv,'r')       %  Just to see a comparison of curves at
                                            %  each iteration, plot it. This statement
                                            % and the next 4 can be commented out to
                                            % prevent plot and pause
        title(['compare curves is = ' num2str(is(k)) ' n = ' num2str(nt(n)) ])
        xlabel('V_ds in Volts')
        ylabel('i_ds in Amps')
        pause(.3)
    end
end
```

```
%  Find point at which the minimum error occurred
[mvtr, mkp] = find(min(min(err)) == err);
ntmin = nt(mvtr(1));  %  MSE index "n"
disp([ 'Estimated n: ' num2str(ntmin) ])
ismin = is(mkp(1));  %  MSE saturation current values
disp([ 'Estimated Is ' num2str(ismin) ])
%  Plot best fit
ids = diodetc(vds,ismin,ntmin,vt);
figure(1)
plot(vds,ids,'g',vds,cv,'r')
title('compare measured data (red) with MSE Shockley Eqn. fit (green)')
xlabel('V_ds in Volts')
ylabel('i_ds in Amps')
```

Best-fit-Curve with Diode Data

Estimated n: 2

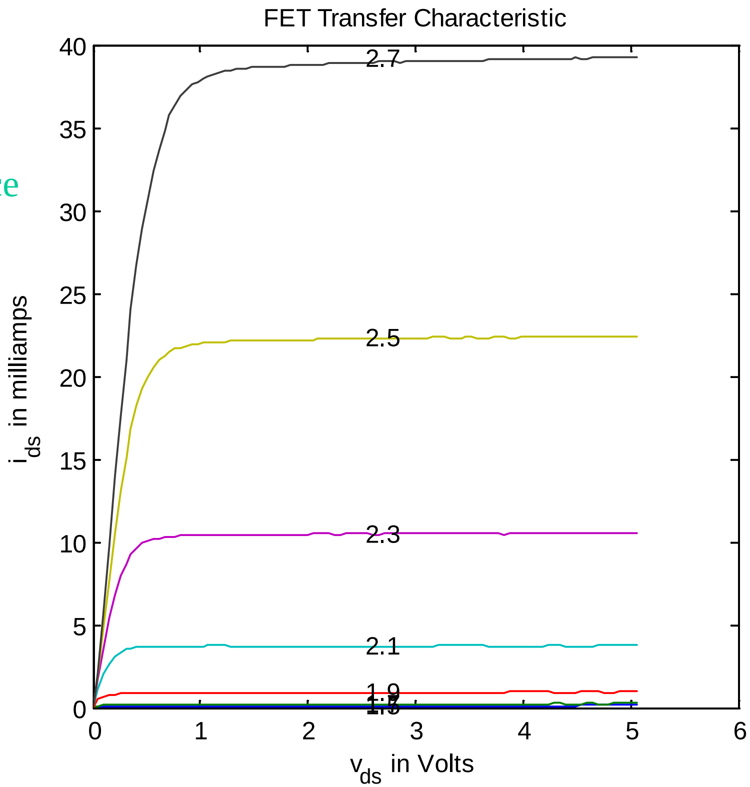
Estimated Is 2.1063e-008



Plot MOSFET TC

Data was collected on the Tektronix's Curve Tracer 370B for a NMOS FET and saved as a CSV file. Open file, process data and plot TC

```
%  Open CSV curve from Curve Tracer Output File
fname = ['E121555E.CSV'];
vstep = [1.496:.2:2.696];  % Need to provide the step sequence
c = getcurves(fname,vstep);
%  Convert Cell Array to vector
[ids, vds, vgs] = interpcurves(c);
% Plot it for observation
figure(1)
plot(vds,ids*1000)
title('FET Transfer Characteristic')
xlabel('v_d_s in Volts')
ylabel('i_d_s in milliamps')
%  Write the Vgs values on the plot near corresponding TC
hold on
% Step through vgs vector and write number on plot
for k=1:length(vgs)
   text(mean(vds),1000*max(ids(k,:)),num2str(vgs(k),2))
end
hold off
```



Curve Fit Parameter Estimation

Matlab Example: Consider a parametric variation of the model for the NMOS FET. Since the function NMOS already implements the NMOS FET I-V curves, use that to write a Matlab script to iterate with a range of values for V_{tr} and K_p for a given V_{gs} while comparing the squared error between the measured i_{DS} values and those predicted with the NMOS function. Find the V_{tr} and K_p that result in the minimum error for several values of V_{gs}

```
%  Extract a curve from Curve Tracer Output
vgs = [1.496:.2:2.696];  %  Gate voltages at which measurement was taken (you need to
                         %  get this information at the time the data is
                         %  collected.  In this case the offset was 1.496, step
                         %  size was .2 volts, and 7 steps were generated.

%  Select row corresponding to the particular vgs value on which to perform
%  curve fit (should be between 1 and length of vgs)
rwtest = 4;
%  Parameters to vary for curve fit.
vto = [.5:.05:3];  %  Range for threshold voltages
kp = [.001:.005:.2];  %  Range of KP values
% Fixed parameters (set W and L to one so they will have no effect)
W= 1;
L=1;
```

```
% read in data and sort in a cell array (rows don't have same number of points so we can't
% use a regular matrix)
c = getcurves(['E121555E.CSV'], vgs);
%  Interpolation so curves and on a regular grid (x-axis)
[m, vds, p] = interpcurves(c);

cv = m(rwtest,:);   %  Get curve from family of curves on which to perform the fit.
vgsv = p(rwtest);   %  Corresponding gate voltage

%  Loop to compute mean squared error for every iteration of test values
%  in parametric function
for n=1:length(vto)  %  Loop through all threshold values
   for k=1:length(kp)  %  Loop through all KP values
      ids = nmos(vds,vgs(rwtest),kp(k),W,L,vto(n));  %  Parametric curve
      err(n,k) = mean(abs(ids - cv).^2);  %  mean square error with measured data
   end
end
```

```
%  Find point at which the minimum error occurred
[mvtr, mkp] = find(min(min(err)) == err);
 kpmin = kp(mkp(1));  %  Estimated kp values
disp([ 'Estimated kp: ' num2str(kpmin) ])
vtrmin = vto(mvtr(1));  %  Estimated threshold gate voltage
disp([ 'Estimated threshold ' num2str(vtrmin) ])
%  Plot best fit
ids = nmos(vds,vgs(rwtest),kpmin,W,L,vtrmin);
figure(1)
plot(vds,ids,'g',vds,cv,'r')
title(['Compare MSE Curve to Data for Vgs = ' num2str(vgs(rwtest))])
xlabel('V_ds in Volts')
ylabel('i_ds in Amps')
figure(2)
%  Look at error surface
imagesc(kp,vto,log10(err))
colormap(jet)
ylabel('Threshold Gate Voltage Values')
xlabel('KP values')
title('Log of MSE Error surface')
colorbar
```

Plot of Best-Fit with Error Surface

Estimated kp: 0.061

Estimated threshold 1.75 Volts