Neuromorphic engineering I

Lab 2: Transistor superthreshold saturation current and drain characteristics

Group number: 4.9

Team member: Huang wencan

Date: 2021.Oc.13

The objective of this lab is to understand *super-threshold* (also called *above-threshold* or *strong inversion*) transistor operation and to understand transistor drain conductance characteristics, particularly *channel length modulation*.

The specific experimental **objectives of this lab** are as follows:

1. To characterize drain current of a transistor as a function of gate voltage in superthreshold operation in the ohmic (triode) and saturation regions.
2. To characterize the drain saturation properties in super-threshold.
3. To characterize drain conductance (the Early effect) and how it scales with transistor length (may not be possible this year) and saturation drain current.

An intuitive and quantitative understanding of all these effects, along with the subthreshold behavior (next week), is useful for the design of effective circuits, especially analog design of high performance amplifiers.

1 Terminology

- above-threshold = super-threshold = strong inversion
- sub-threshold = below-threshold = weak inversion
- triode region = ohmic region = linear drain conductance behavior with small drain-source voltage
- saturation = large $V_{\rm ds}$
- overdrive = $V_{\rm g}-V_{\rm T}$
- $U_{\rm T} = kT/q$ = thermal voltage = 25mV at room temperature
- $V_{\rm T}$ = threshold voltage = 0.4V to 0.8V depending on process

2 Useful Quantities

The following is a list of the physical parameters and constants we will be referring to in this lab, along with their values when appropriate. The units that are most natural for these quantities are also included; these units are not self--consistent, so make sure you convert the units when appropriate.

ϵ_0 : Permittivity of vacuum = $8.86\times10^{-12} \rm{ F/m}$

ϵ_{Si} : Relative permittivity of ${\rm Si}$ = $11.7\epsilon_0$

ϵ_{ox} : Relative permittivity of ${\rm SiO_2}$ = $3.9\epsilon_0$

μ_n : electron surface mobility, ${\rm cm^2/V/s}$

μ_p : hole surface mobility, ${\rm cm^2/V/s}$

C_{ox} : gate capacitance across the oxide per unit area, ${\rm fF/\mu m^{2}}$

C_{dep} : capacitance of depletion region per unit area, ${\rm fF/\mu m^{2}}$

t_{ox} : gate oxide thickness $\approx$ 3.8 nm for the class chip in 180 nm techology.

V_T : threshold voltage, V (V_{T0} is V_T when V_s = 0).

W : electrical width of transistor channel, $= 4~\mu$m for both devices in this lab

L : electrical length of transistor channel, $= 4~\mu$m for both devices in this lab

β $\equiv \mu C_{ox} W/L,\;\mu{\rm A/V^2}$

V_E : Early voltage, characterizes drain conductance.

3 Prelab

Write the expressions/eqations in LaTeX, like Vod = $V_{\rm g}-V_{\rm T}$, or upload the pictures of handwritten expressions.

- For nFET, write the most general expression for I_{ds} above threshold in terms of $V_g,\;V_s,\;V_d$ (all voltages are referenced to the bulk), and the parameters and constants given above. Leave out the drain conductance Early effect in this equation. Assume $\kappa=1$ and that $V_{Tn}>0$.

The most general expression for I_{ds} is $I_{ds}=\mu\rho\xi$
When $V_{d}-V_{s}$ is small, I_{ds} = $\beta (V_{gs}-V_{Tn})V_{ds}$=$\beta(V_{g}-V_{s}-V_{Tn})(V_{d}-V_{s})$
When $V_{d}-V_{s}$ is large (>4V_{T}), I_{ds} = $\frac{\beta}{2}(V_{g}-V_{s}-V_{Tn})^{2}$

- For pFET, write the most general expression for I_{ds} above threshold in terms of $V_g,\;V_s,\;V_d$ (all voltages are referenced to the bulk), and the parameters and constants given above. Leave out the drain conductance Early effect in this equation. Assume $\kappa=1$ and that $V_{Tp}<0$.

The most general expression for I_{ds} is $I_{ds}=\mu\rho\xi$
When $V_{d}-V_{s}$ is small, I_{ds} = $\beta (V_{gs}-V_{Tp})V_{ds}$=$\beta(V_{g}-V_{s}-V_{Tp})(V_{d}-V_{s})$
When $V_{d}-V_{s}$ is large (<-4V_{T}), I_{ds} = $\frac{\beta}{2}(V_{g}-V_{s}-V_{Tp})^{2}$

- For nFET, sketch graphs of I_{ds} vs the V_d for several gate voltages V_g above threshold, with $V_s=0$. Indicate the ohmic and saturation regions and the behavior of the saturation voltage V_{dsat} as the gate overdrive voltage increases.

- For nFET, derive an expression for the current I_{ds} in the ohmic region in terms of V_g and $V_{ds} \equiv V_d-V_s,$. You may assume that $V_s=0.$ Sketch a graph of I_{ds} vs $V_g,$ showing V_{T0} and an expression for the slope.

For ohmic region, I_{ds} = $\beta (V_{gs}-V_{T0})V_{ds}$=$\beta(V_{g}-V_{T0})V_{ds}$

- For nFET, state the drain voltage condition for above-threshold saturation and derive an expression for the saturation current I_{dsat} in terms of V_g. Sketch a graph of $\sqrt{I_{dsat}}$ vs V_g with $V_s=0,$ showing V_{T0} and an expression for the slope. Do not consider the Early effect here.

For saturation, it requires $V_{ds}>4V_{T}=0.1mV$
$I_{dsat} = \frac{\beta}{2}(V_{g}-V_{T0})^{2}$

- Calculate C_{ox} for the classchip from the values given above. What is C_{ox} per square micron in fF?

According to the definition, per square micron $C_{ox} = \epsilon_{ox}\frac{A}{d} = 3.9\times 8.86\times 10^{-12}F/m\times\frac{10^{-12}m^{2}}{3.8\times 10^{-9}m}= 9.09fF$

- Write the expression for the drain current in saturation including the Early effect, using I_{dsat} to represent the saturation current in the absence of the Early effect. Use V_E to represent the Early voltage.

$I = I_{dsat} (1 + \frac{V_{d}}{V_{E}})$

4 Setup

4.1 Connect the device

In [2]:

```
# import the necessary library to communicate with the hardware
#import sys
#sys.path.append('/home/junren/software/CoACH_Teensy_interface/build/pc/pyplane'

import pyplane
import time
```

In [3]:

```
# create a Plane object and open the communication
if 'p' not in locals():
    p = pyplane.Plane()
    try:
        p.open('/dev/ttyACM0') # Open the USB device ttyACM0 (the board).
    except RuntimeError as e:
        print(e)
```

```
# Note that if you plug out and plug in the USB device in a short time interval,
# then you may get error messages with open(...ttyACM0). So please avoid frenque
```

In [4]:

```
p.get_firmware_version()   #firmware version should be 1.8.3
```

Out[4]:

```
(1, 8, 3)
```

In [5]:

```
# Send a reset signal to the board, check if the LED blinks
p.reset(pyplane.ResetType.Soft)

time.sleep(1)
# NOTE: You must send this request events every time you do a reset operetion, c
# Because the class chip need to do handshake to get the communication correct.
p.request_events(1)
```

In [6]:

```
# Try to read something, make sure the chip responses
p.read_current(pyplane.AdcChannel.GO0_N)
```

Out[6]:

```
1.530761721824092e-07
```

In [6]:

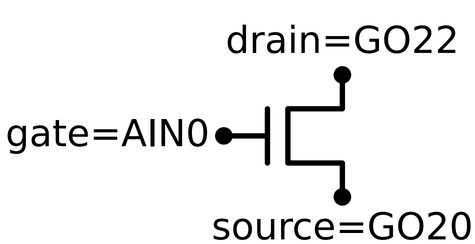
```
# If any of the above steps fail, delete the object, and restart the kernel

# del p
```

4.2 Configurations for N-FET

In [8]:

```
# uses schemdraw, you may have to install it in order to run it on your PC
import schemdraw
import schemdraw.elements as elm
d = schemdraw.Drawing()
Q = d.add(elm.NFet, reverse=True)
d.add(elm.Dot, xy=Q.gate, lftlabel='gate=AIN0')
d.add(elm.Dot, xy=Q.drain, toplabel='drain=GO22')
d.add(elm.Dot, xy=Q.source, botlabel='source=GO20')
d.draw()
```

Out[8]:



To cancel out the leakage current and shunt resistance, you may need to do a subtraction in Section 5.1.

$I_{ds} = I_{GO20} - I_{GO20}|_{V_{gs} = 0}$

Note: It's better to measure source because its leakage is constant in this lab

- You have to set the input voltage demultiplexer by sending a configuration event:

In [8]:
```
# Configure NFET, set the input voltage demultiplexer by AER event.
# Note selectlines we should choose for the NFET
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine2, \
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]

p.send_coach_events(events)
```

- Check the configuration is correct. If the measured result is not as expected, try sending the configration event again.

In [9]:
```
# set source voltage
vs = 0.0
Vs_monitor = p.set_voltage(pyplane.DacChannel.GO20,vs)
print("The source voltage is set to {} V".format(Vs_monitor))
```

```
The source voltage is set to 0.0 V
```

In [10]:
```
# set drain voltage
vd = 0.0
Vd_monitor =p.set_voltage(pyplane.DacChannel.GO22,vd)
print("The drain voltage is set to {} V".format(Vd_monitor))
```

```
The drain voltage is set to 0.0 V
```

In [11]:
```
# set gate voltage
vg = 0.0
Vg_monitor = p.set_voltage(pyplane.DacChannel.AIN0, vg)
print("The gate voltage is set to {} V".format(Vg_monitor))
```

```
The gate voltage is set to 0.0 V
```

In [12]:
```
# read I_{ds}
I_s = p.read_current(pyplane.AdcChannel.GO20_N)      #source
print("The measured source current is {} A".format(I_s))

time.sleep(0.1)  # wait for it to settle

I_d = p.read_current(pyplane.AdcChannel.GO22)       #drain
print("The measured drain current is {} A".format(I_d))
```

```
The measured source current is 6.347656267280399e-07 A
The measured drain current is 9.033203127728484e-07 A
```

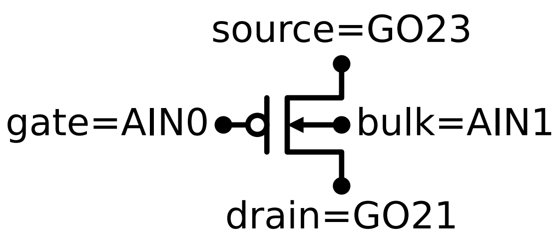
- Question: Check if the measured currents change with different gate voltages?

4.3 Configurations for P-FET

In [13]:

```
# uses schemdraw, you may have to install it in order to run it on your PC
import schemdraw
import schemdraw.elements as elm
d = schemdraw.Drawing()
Q = d.add(elm.PFet, reverse=True, bulk=True)
d.add(elm.Dot, xy=Q.gate, lftlabel='gate=AIN0')
d.add(elm.Dot, xy=Q.bulk, rgtlabel='bulk=AIN1')
d.add(elm.Dot, xy=Q.drain, botlabel='drain=GO21')
d.add(elm.Dot, xy=Q.source, toplabel='source=GO23')
d.draw()
```

Out[13]:



Hint: To cancel out the leakage current and shunt resistance, you may need to do a subtraction:

$I_{ds} = I_{GO23} - I_{GO23}|_{V_{gs} = 0} $

Note: Measure drain of PFET in this lab. Also think about the difference of V_{gs} between PMOS and NMOS?

- You have to choose the input voltage demultiplexer by sending a configuration event (make sure LED1 blinks):

In [14]:

```
# Configure PFET, set the input voltage demultiplexer by AER event.
# Note selectlines we should choose for the PFET
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine1, \
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]

p.send_coach_events(events)
```

- Check the configuration is correct. If the measured result is not as expected, try sending the event again.

In [15]:

```
# set trial voltages
# set bulk voltage
p.set_voltage(pyplane.DacChannel.AIN1, 0.0)
Vb_p = p.get_set_voltage(pyplane.DacChannel.AIN1)
print("The bulk voltage is set to {} V".format(Vb_p))
time.sleep(0.1)  # wait 0.1s for it to settle

# set source voltage
p.set_voltage(pyplane.DacChannel.GO23, 0.0)
```

```
Vs_p = p.get_set_voltage(pyplane.DacChannel.GO23)
print("The source voltage is set to {} V".format(Vs_p))
time.sleep(0.1)  # wait 0.1s for it to settle

# set drain voltage
p.set_voltage(pyplane.DacChannel.GO21, 0.0)
Vd_p = p.get_set_voltage(pyplane.DacChannel.GO21)
print("The drain voltage is set to {} V".format(Vd_p))
time.sleep(0.1)  # wait for it to settle

# set gate voltage
p.set_voltage(pyplane.DacChannel.AIN0, 0.0)
Vg_p = p.get_set_voltage(pyplane.DacChannel.AIN0)
print("The gate voltage is set to {} V".format(Vg_p))
```

```
The bulk voltage is set to 0.0 V
The source voltage is set to 0.0 V
The drain voltage is set to 0.0 V
The gate voltage is set to 0.0 V
```

In [16]:

```
# read I_{ds}
Is_p = p.read_current(pyplane.AdcChannel.GO21_N)
print("The measured source current of PMOS is {} A".format(Is_p))

time.sleep(0.1)  # wait for it to settle

Id_p = p.read_current(pyplane.AdcChannel.GO23)
print("The measured drain current of PMOS is {} A".format(Id_p))
```

```
The measured source current of PMOS is 4.39453117451194e-07 A
The measured drain current of PMOS is 3.906249901319825e-07 A
```

5 Ohmic region

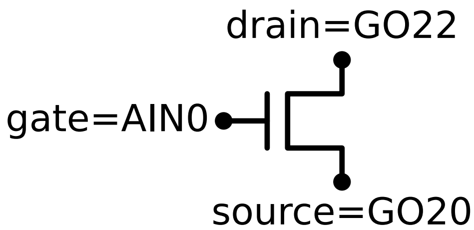
In this experiment you will characterize the *linear* dependence of the current on the gate voltage in the strong-inversion ohmic region.

5.1 N-FET

In [9]:

```
# uses schemdraw, you may have to install it in order to run it on your PC
import schemdraw
import schemdraw.elements as elm
d = schemdraw.Drawing()
Q = d.add(elm.NFet, reverse=True)
d.add(elm.Dot, xy=Q.gate, lftlabel='gate=AIN0')
d.add(elm.Dot, xy=Q.drain, toplabel='drain=GO22')
d.add(elm.Dot, xy=Q.source, botlabel='source=GO20')
d.draw()
```

Out[9]:



(a) Configure the chip following Section 4.2 if you haven't

(b) Measure I_{ds} as a function of V_g in ohmic region

In [10]:
```
# Configure NFET, set the input voltage demultiplexer by AER event.
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine2, \
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]

p.send_coach_events(events)
```

- What will be the fixed value for source and drain voltages?

Answer:

In [11]:
```
# set source voltage
vs = 0.0
Vs_monitor = p.set_voltage(pyplane.DacChannel.GO20,vs)
print("The source voltage is set to {} V".format(Vs_monitor))
# set drain voltage
```

```
The source voltage is set to 0.0 V
```

In [12]:
```
# set drain voltage
vd = 0.08
Vd_monitor =p.set_voltage(pyplane.DacChannel.GO22,vd)
print("The drain voltage is set to {} V".format(Vd_monitor))
```

```
The drain voltage is set to 0.07917889207601547 V
```

- For very close voltages, you may want to call `get_set_voltage` to check the actual output of the DAC.

In [13]:
```
# get set voltage
Vs_n = p.get_set_voltage(pyplane.DacChannel.GO20)
print("The source voltage is set to {} V".format(Vs_n))

time.sleep(0.1)  # wait for it to settle

# get set voltage
Vd_n = p.get_set_voltage(pyplane.DacChannel.GO22)
print("The drain voltage is set to {} V".format(Vd_n))
```

```
The source voltage is set to 0.0 V
```

```
The drain voltage is set to 0.07917889207601547 V
```

- Data aquisition

In [14]:
```
# sweep gate voltage
import time
import numpy as np

# Get the leakage current, Read Ids=Ids0 at Vg = 0
p.set_voltage(pyplane.DacChannel.AIN0,0.0)
time.sleep(0.5) # wait 0.5 second for it to settle
Is0_n =  p.read_current(pyplane.AdcChannel.GO20_N)
print("Offset Is0_n: {} A".format(Is0_n))

number = 50
Vg_sets = np.linspace(0,1.8,num = number).reshape(-1,1)
Is_sets = np.zeros(number).reshape(-1,1)

for n in range(number):
    # set gate voltage
    p.set_voltage(pyplane.DacChannel.AIN0,Vg_sets[n])
    vg_set = p.get_set_voltage(pyplane.DacChannel.AIN0)
    print("The gate voltage is set to {} V".format(vg_set))   ## print the gate
    time.sleep(0.05)  # wait for it to settle

    # read I_{ds}
    Is_set = p.read_current(pyplane.AdcChannel.GO20_N)
    print("The measured source current is {} A".format(Is_set))  ## print the ra

    # substract leakage current
    Is_sets[n] = Is_set - Is0_n
```

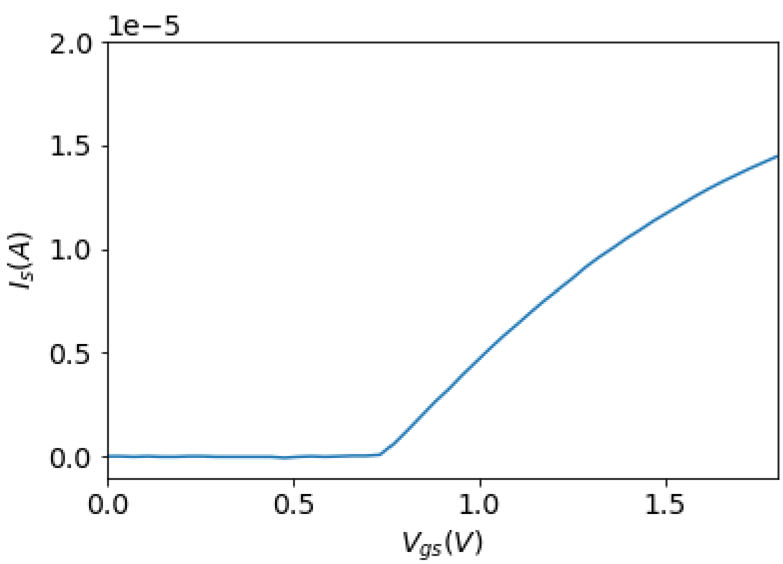
```
Offset Is0_n: 6.347656267280399e-07 A
The gate voltage is set to 0.0 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.03519061952829361 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.07214076817035675 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.10909092426300049 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.14604108035564423 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.18299122154712677 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.2199413776397705 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.25689151883125305 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.293841689825058 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.329032301902771 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.36598244309425354 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.4029325842857361 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.439882755279541 V
```

```
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.47683289647102356 V
The measured source current is 5.615234499600774e-07 A
The gate voltage is set to 0.5137830376625061 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.550733208656311 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.587683379650116 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.6228739619255066 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.6598241329193115 V
The measured source current is 6.591797045985004e-07 A
The gate voltage is set to 0.6967743039131165 V
The measured source current is 6.591797045985004e-07 A
The gate voltage is set to 0.7337244153022766 V
The measured source current is 7.080078034960025e-07 A
The gate voltage is set to 0.7706745862960815 V
The measured source current is 1.2451172324290383e-06 A
The gate voltage is set to 0.8076247572898865 V
The measured source current is 1.904296823340701e-06 A
The gate voltage is set to 0.8445748686790466 V
The measured source current is 2.587890548966243e-06 A
The gate voltage is set to 0.8815250396728516 V
The measured source current is 3.271484274591785e-06 A
The gate voltage is set to 0.9167156219482422 V
The measured source current is 3.8818361645098776e-06 A
The gate voltage is set to 0.9536657929420471 V
The measured source current is 4.565429662761744e-06 A
The gate voltage is set to 0.990615963935852 V
The measured source current is 5.2001951189595275e-06 A
The gate voltage is set to 1.0275660753250122 V
The measured source current is 5.834961029904662e-06 A
The gate voltage is set to 1.064516305923462 V
The measured source current is 6.4453124650754035e-06 A
The gate voltage is set to 1.101466417312622 V
The measured source current is 7.006835858192062e-06 A
The gate voltage is set to 1.1384165287017822 V
The measured source current is 7.592773272335762e-06 A
The gate voltage is set to 1.175366759300232 V
The measured source current is 8.15429666545242e-06 A
The gate voltage is set to 1.2105573415756226 V
The measured source current is 8.666992471262347e-06 A
The gate voltage is set to 1.2475074529647827 V
The measured source current is 9.179687367577571e-06 A
The gate voltage is set to 1.2844576835632324 V
The measured source current is 9.741211215441581e-06 A
The gate voltage is set to 1.3214077949523926 V
The measured source current is 1.0229492545477115e-05 A
The gate voltage is set to 1.3583579063415527 V
The measured source current is 1.0668944923963863e-05 A
The gate voltage is set to 1.3953081369400024 V
The measured source current is 1.1132812687719706e-05 A
The gate voltage is set to 1.4322582483291626 V
The measured source current is 1.1547851499926765e-05 A
The gate voltage is set to 1.4692083597183228 V
The measured source current is 1.1987304787908215e-05 A
The gate voltage is set to 1.504399061203003 V
The measured source current is 1.2377930033835582e-05 A
The gate voltage is set to 1.541349172592163 V
The measured source current is 1.2768554370268248e-05 A
```

```
The gate voltage is set to 1.5782992839813232 V
The measured source current is 1.3159179616195615e-05 A
The gate voltage is set to 1.615249514579773 V
The measured source current is 1.352539038634859e-05 A
The gate voltage is set to 1.652199625968933 V
The measured source current is 1.3867187590221874e-05 A
The gate voltage is set to 1.6891497373580933 V
The measured source current is 1.4184570318320766e-05 A
The gate voltage is set to 1.726099967956543 V
The measured source current is 1.4501953046419658e-05 A
The gate voltage is set to 1.7630500793457031 V
The measured source current is 1.4794922208238859e-05 A
The gate voltage is set to 1.7982406616210938 V
The measured source current is 1.5087890460563358e-05 A
```

In [15]:

```
# plot
import matplotlib.pyplot as plt
plt.plot(Vg_sets,Is_sets)
plt.ylim([-0.1e-5,2e-5])
plt.xlim([0,1.8])
plt.xlabel(r'$V_{gs} (V)$')
plt.ylabel(r'$I_{s} (A)$')
plt.show()
```



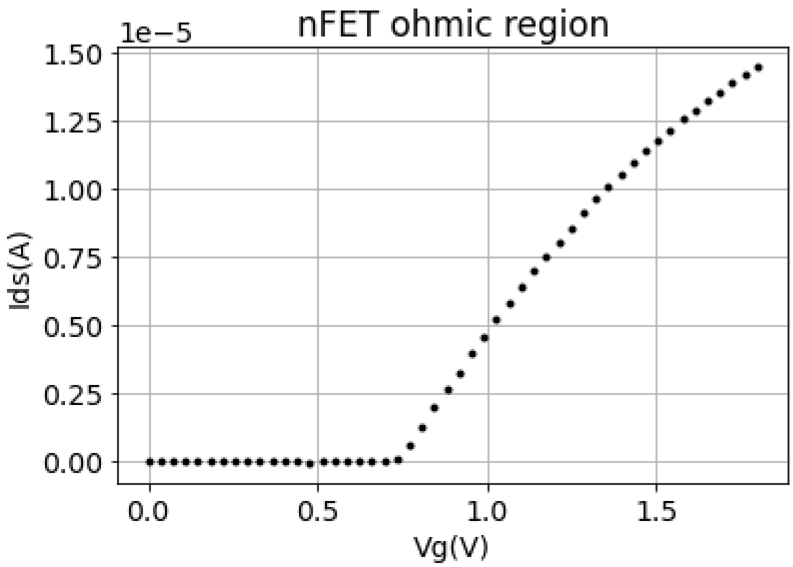
In [16]:

```
# if the data looks nice, save it!
Lab2_data_nFETVgIds_Omic = [Vg_sets.reshape(number),Is_sets.reshape(number)]
np.savetxt('./data/Lab2_data_nFETVgIds.csv', Lab2_data_nFETVgIds_Omic, delimite
# example :
# Lab2_data_nFETVgIds_Omic = [Vg_n,Is_n]
# save to csv file
# np.savetxt('./data/Lab2_data_nFETVgIds.csv', Lab2_data_nFETVgIds_Omic, delimit
```

In [60]:

```
# Load data you saved and plot, to check if the data is saved correctly or not
import numpy as np
import matplotlib.pyplot as plt
Vgn_save, Isn_save = np.loadtxt('./data/Lab2_data_nFETVgIds.csv',delimiter=",")
plt.rcParams.update({'font.size': 14})
plt.title('nFET ohmic region')
plt.plot(Vgn_save, Isn_save, '.k')
```

```
plt.xlabel('Vg(V)')
plt.ylabel('Ids(A)')
plt.grid()
plt.show()

# example :
# Vgn_save, Isn_save = np.loadtxt('./data/Lab2_data_nFETVgIds.csv',delimiter=",'
#plt.rcParams.update({'font.size': 14})
#plt.plot(Vgn_save, Isn_save, '.k')
#plt.xlabel('Vg(V)')
#plt.ylabel('Ids(A)')
#plt.grid()
#plt.show()
```



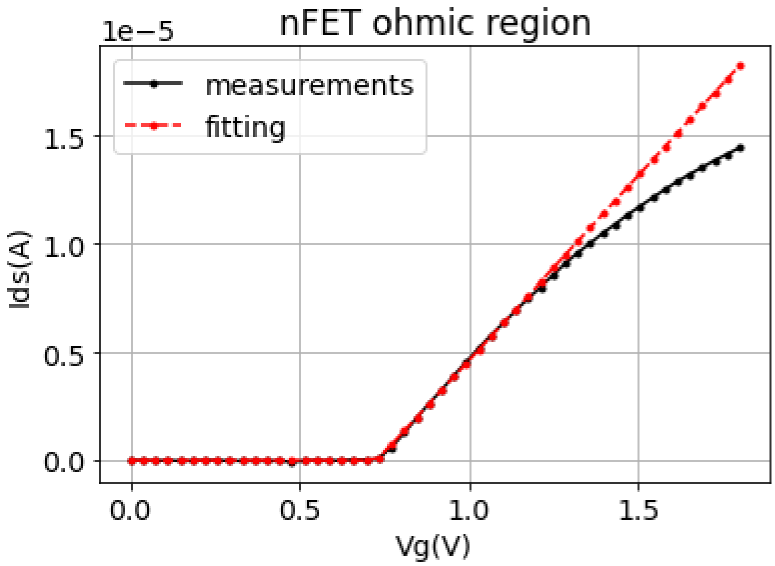
In [61]:

```
# extract the valid range
# the range choosen it between 0.8 and 1.2
Vg_clip = Vgn_save[(Vgn_save> 0.8) & (Vgn_save < 1.2)]
Ids_clip = Isn_save[(Vgn_save> 0.8) & (Vgn_save < 1.2)]
```

In [62]:

```
# fit in the valid range (you may want to go back and add the fitted line in the
fit = np.polyfit(Vg_clip,Ids_clip,1)
print('slope:',fit[0])
plt.rcParams.update({'font.size': 14})
plt.title('nFET ohmic region')
plt.plot(Vgn_save, Isn_save, '.k-')
plt.plot(Vgn_save, np.clip(Vgn_save*fit[0]+fit[1],a_min=0,a_max = 1),'.r--')
plt.xlabel('Vg(V)')
plt.ylabel('Ids(A)')
plt.legend(['measurements','fitting'])
plt.grid()
plt.show()
```

```
slope: 1.7044097840062937e-05
```



(c) Determine V_{T0} and β for both devices by fitting your data to the expression derived in the prelab

In [63]:

```
# V_T0
v_t0n = -fit[1]/fit[0]
print('V_T0 =',v_t0n, 'V')
```

```
V_T0 = 0.728013656175449 V
```

In [64]:

```
# beta => m/Vd
vd = 0.08
betan = fit[0]/vd

print('beta_n =',betan*1e6, 'uA/V/V')
```

```
beta_n = 213.0512230007867 uA/V/V
```

5.2 P-FET

(a) Configure the chip following Section 4.3 if you haven't

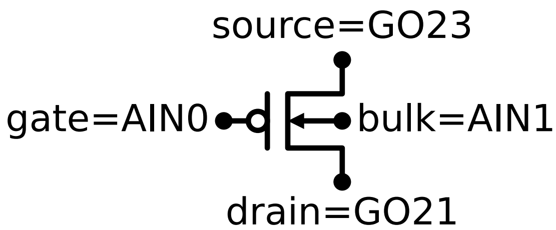
(b) Measure I_{ds} as a function of V_g in ohmic region

- What will be the fixed value for bulk, source and drain voltages?

In [17]:

```
# uses schemdraw, you may have to install it in order to run it on your PC
import schemdraw
import schemdraw.elements as elm
d = schemdraw.Drawing()
Q = d.add(elm.PFet, reverse=True, bulk=True)
d.add(elm.Dot, xy=Q.gate, lftlabel='gate=AIN0')
d.add(elm.Dot, xy=Q.bulk, rgtlabel='bulk=AIN1')
d.add(elm.Dot, xy=Q.drain, botlabel='drain=GO21')
d.add(elm.Dot, xy=Q.source, toplabel='source=GO23')
d.draw()
```

Out[17]:



In [18]:

```
# Configure PFET, set the input voltage demultiplexer by AER event.
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine1, \
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]

p.send_coach_events(events)
```

In [19]:

```
# set bulk voltage
vb = 1.8
Vb_monitor = p.set_voltage(pyplane.DacChannel.AIN1,vb)
print("The bulk voltage is set to {} V".format(Vb_monitor))
time.sleep(0.05)  # wait for it to settle

# set source voltage
vs = 1.8
Vs_monitor = p.set_voltage(pyplane.DacChannel.GO23,vs)
print("The source voltage is set to {} V".format(Vs_monitor))
time.sleep(0.05)  # wait for it to settle

# set drain voltage
vd = 1.72
Vd_monitor = p.set_voltage(pyplane.DacChannel.GO21,vd)
print("The drain voltage is set to {} V".format(Vd_monitor))
time.sleep(0.05)  # wait for it to settle

# Print I_ds for checking
I_ds =  p.read_current(pyplane.AdcChannel.GO21_N)
print("Offset I_ds: {} A".format(I_ds))
```

```
The bulk voltage is set to 1.7982406616210938 V
The source voltage is set to 1.7982406616210938 V
The drain voltage is set to 1.7190617322921753 V
Offset I_ds: 6.665039109066129e-06 A
```

- For very close voltages, you may want to call `get_set_voltage` to check the actual output of the DAC.

In [20]:

```
# get set voltage
Vb_monitor = p.get_set_voltage(pyplane.DacChannel.AIN1)
print("The bulk voltage is set to {} V".format(Vb_monitor))
Vd_monitor = p.get_set_voltage(pyplane.DacChannel.GO21)
print("The drain voltage is set to {} V".format(Vd_monitor))
Vs_monitor = p.get_set_voltage(pyplane.DacChannel.GO23)
print("The source voltage is set to {} V".format(Vs_monitor))
```

```
The bulk voltage is set to 1.7982406616210938 V
The drain voltage is set to 1.7190617322921753 V
The source voltage is set to 1.7982406616210938 V
```

- Data aquisition

In [21]:

```
# sweep gate voltage
import time
import numpy as np

# Get the leakage current, Read Ids=Ids0 at Vg = 1.8
p.set_voltage(pyplane.DacChannel.AIN0,1.8)
time.sleep(0.5) # wait 0.5 second for it to settle
Id0_p =  p.read_current(pyplane.AdcChannel.GO21_N)
Is0_p =  p.read_current(pyplane.AdcChannel.GO23)
print("Offset Id0_p: {} A".format(Id0_p))
print("Offset Is0_p: {} A".format(Is0_p))

number = 50
Vg_sets = np.linspace(1.8,0.0,num = number).reshape(-1,1)
Id_sets = np.zeros(number).reshape(-1,1)
Is_sets = np.zeros(number).reshape(-1,1)
for n in range(number):
    # set gate voltage
    p.set_voltage(pyplane.DacChannel.AIN0,Vg_sets[n])
    vg_set = p.get_set_voltage(pyplane.DacChannel.AIN0)
    print("The gate voltage is set to {} V".format(vg_set))   ## print the gate
    time.sleep(0.2)  # wait for it to settle

    # read I_{ds}
    Id_set = p.read_current(pyplane.AdcChannel.GO21_N)
    Is_set = p.read_current(pyplane.AdcChannel.GO23)
    print("The measured drain current is {} A".format(Id_set))  ## print the raw
    print("The measured source current is {} A".format(Is_set))  ## print the ra

    # substract leakage current
    Id_sets[n] = Id_set - Id0_p
    Is_sets[n] = Is_set - Is0_p
```

```
Offset Id0_p: 6.054687673895387e-06 A
Offset Is0_p: 6.372070401994279e-06 A
The gate voltage is set to 1.7982406616210938 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.7630500793457031 V
The measured drain current is 6.054687673895387e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.726099967956543 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.6891497373580933 V
The measured drain current is 6.079101694922429e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.652199625968933 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.615249514579773 V
The measured drain current is 6.127929736976512e-06 A
The measured source current is 6.372070401994279e-06 A
```

```
The gate voltage is set to 1.5782992839813232 V
The measured drain current is 6.079101694922429e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.541349172592163 V
The measured drain current is 6.079101694922429e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.504399061203003 V
The measured drain current is 6.079101694922429e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.4692083597183228 V
The measured drain current is 6.176757779030595e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.4322582483291626 V
The measured drain current is 6.176757779030595e-06 A
The measured source current is 6.347656380967237e-06 A
The gate voltage is set to 1.3953081369400024 V
The measured drain current is 6.079101694922429e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.3583579063415527 V
The measured drain current is 6.079101694922429e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.3214077949523926 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.2844576835632324 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.39648442302132e-06 A
The gate voltage is set to 1.2475074529647827 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.372070401994279e-06 A
The gate voltage is set to 1.2105573415756226 V
The measured drain current is 6.10351571594947e-06 A
The measured source current is 6.39648442302132e-06 A
The gate voltage is set to 1.175366759300232 V
The measured drain current is 6.127929736976512e-06 A
The measured source current is 6.39648442302132e-06 A
The gate voltage is set to 1.1384165287017822 V
The measured drain current is 6.127929736976512e-06 A
The measured source current is 6.39648442302132e-06 A
The gate voltage is set to 1.101466417312622 V
The measured drain current is 6.152343758003553e-06 A
The measured source current is 6.420898444048362e-06 A
The gate voltage is set to 1.064516305923462 V
The measured drain current is 6.152343758003553e-06 A
The measured source current is 6.420898444048362e-06 A
The gate voltage is set to 1.0275660753250122 V
The measured drain current is 6.152343758003553e-06 A
The measured source current is 6.4453124650754035e-06 A
The gate voltage is set to 0.990615963935852 V
The measured drain current is 6.176757779030595e-06 A
The measured source current is 6.4453124650754035e-06 A
The gate voltage is set to 0.9536657929420471 V
The measured drain current is 6.176757779030595e-06 A
The measured source current is 6.469726486102445e-06 A
The gate voltage is set to 0.9167156219482422 V
The measured drain current is 6.2011718000576366e-06 A
The measured source current is 6.469726486102445e-06 A
The gate voltage is set to 0.8815250396728516 V
The measured drain current is 6.274413863138761e-06 A
The measured source current is 6.469726486102445e-06 A
The gate voltage is set to 0.8445748686790466 V
```

```
The measured drain current is 6.274413863138761e-06 A
The measured source current is 6.494140507129487e-06 A
The gate voltage is set to 0.8076247572898865 V
The measured drain current is 6.274413863138761e-06 A
The measured source current is 6.518554528156528e-06 A
The gate voltage is set to 0.7706745862960815 V
The measured drain current is 6.298828338913154e-06 A
The measured source current is 6.518554528156528e-06 A
The gate voltage is set to 0.7337244153022766 V
The measured drain current is 6.274413863138761e-06 A
The measured source current is 6.567383024957962e-06 A
The gate voltage is set to 0.6967743039131165 V
The measured drain current is 6.3232423599401955e-06 A
The measured source current is 6.54296854918357e-06 A
The gate voltage is set to 0.6598241329193115 V
The measured drain current is 6.347656380967237e-06 A
The measured source current is 6.567383024957962e-06 A
The gate voltage is set to 0.6228739619255066 V
The measured drain current is 6.347656380967237e-06 A
The measured source current is 6.591797045985004e-06 A
The gate voltage is set to 0.587683379650116 V
The measured drain current is 6.372070401994279e-06 A
The measured source current is 6.6162110670120455e-06 A
The gate voltage is set to 0.550733208656311 V
The measured drain current is 6.372070401994279e-06 A
The measured source current is 6.640625088039087e-06 A
The gate voltage is set to 0.5137830376625061 V
The measured drain current is 6.39648442302132e-06 A
The measured source current is 6.640625088039087e-06 A
The gate voltage is set to 0.47683289647102356 V
The measured drain current is 6.39648442302132e-06 A
The measured source current is 6.665039109066129e-06 A
The gate voltage is set to 0.439882755279541 V
The measured drain current is 6.494140507129487e-06 A
The measured source current is 6.68945313009317e-06 A
The gate voltage is set to 0.4029325842857361 V
The measured drain current is 6.4453124650754035e-06 A
The measured source current is 6.665039109066129e-06 A
The gate voltage is set to 0.36598244309425354 V
The measured drain current is 6.4453124650754035e-06 A
The measured source current is 6.68945313009317e-06 A
The gate voltage is set to 0.329032301902771 V
The measured drain current is 6.469726486102445e-06 A
The measured source current is 6.713867151120212e-06 A
The gate voltage is set to 0.293841689825058 V
The measured drain current is 6.469726486102445e-06 A
The measured source current is 6.713867151120212e-06 A
The gate voltage is set to 0.25689151883125305 V
The measured drain current is 6.494140507129487e-06 A
The measured source current is 6.7382811721472535e-06 A
The gate voltage is set to 0.2199413776397705 V
The measured drain current is 6.494140507129487e-06 A
The measured source current is 6.7382811721472535e-06 A
The gate voltage is set to 0.18299122154712677 V
The measured drain current is 6.6162110670120455e-06 A
The measured source current is 6.762695193174295e-06 A
The gate voltage is set to 0.14604108035564423 V
The measured drain current is 6.54296854918357e-06 A
The measured source current is 6.762695193174295e-06 A
The gate voltage is set to 0.10909092426300049 V
The measured drain current is 6.567383024957962e-06 A
```

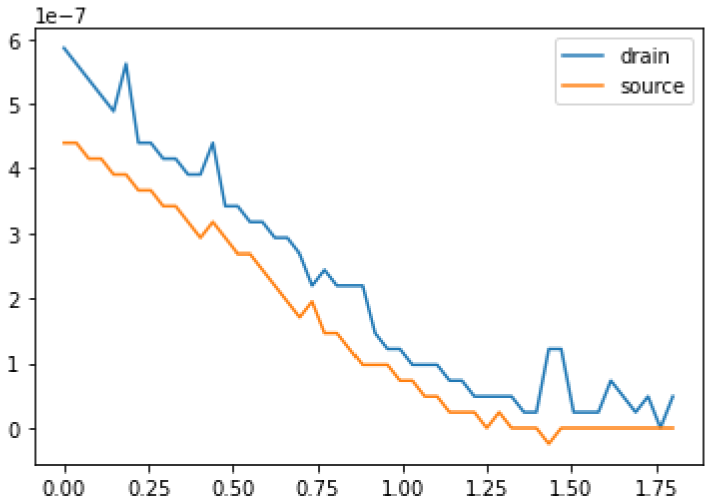
```
The measured source current is 6.787109214201337e-06 A
The gate voltage is set to 0.07214076817035675 V
The measured drain current is 6.591797045985004e-06 A
The measured source current is 6.787109214201337e-06 A
The gate voltage is set to 0.03519061952829361 V
The measured drain current is 6.6162110670120455e-06 A
The measured source current is 6.811523235228378e-06 A
The gate voltage is set to 0.0 V
The measured drain current is 6.640625088039087e-06 A
The measured source current is 6.811523235228378e-06 A
```

In [22]:

```
# plot
import matplotlib.pyplot as plt
plt.plot(Vg_sets,Id_sets)
plt.plot(Vg_sets,Is_sets)
plt.legend(['drain','source'])
```

Out[22]:

```
<matplotlib.legend.Legend at 0x7f12f6df5af0>
```

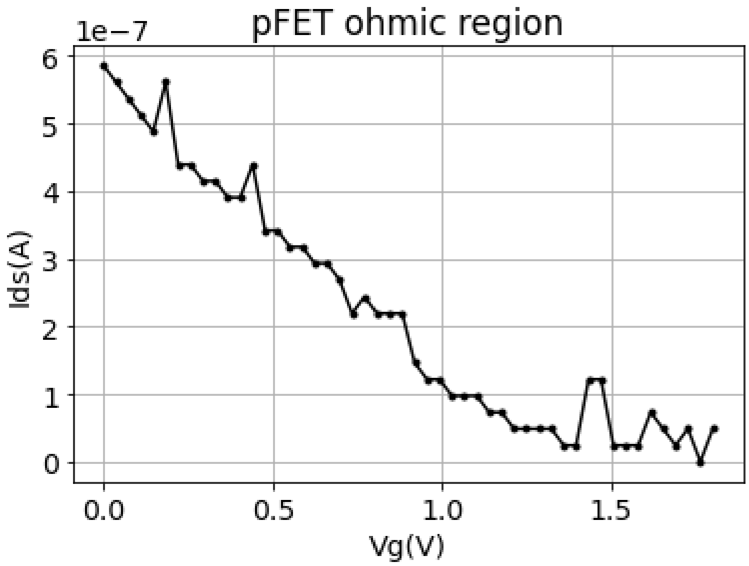


In [23]:

```
# if the data looks nice, save it!
Lab2_data_pFETVgIds_Omic = [Vg_sets.reshape(number),Id_sets.reshape(number)]
np.savetxt('./data/Lab2_data_pFETVgIds.csv', Lab2_data_pFETVgIds_Omic, delimiter
```

In [65]:

```
# Load data you saved and plot, to check if the data is saved correctly or not
import numpy as np
import matplotlib.pyplot as plt
Vgp_save, Idp_save = np.loadtxt('./data/Lab2_data_pFETVgIds.csv',delimiter=",")
plt.title('pFET ohmic region')
plt.rcParams.update({'font.size': 14})
plt.plot(Vgp_save, Idp_save, '.k-')
plt.xlabel('Vg(V)')
plt.ylabel('Ids(A)')
plt.grid()
plt.show()
```



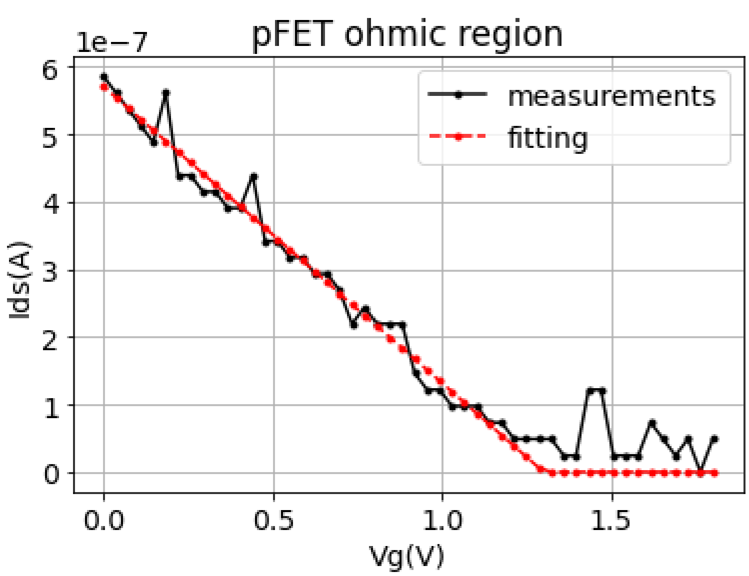
In [66]:

```
# extract the valid range
Vg_clip = Vgp_save[(Vgp_save<1.2)&(Vgp_save>0.3)]
Ids_clip = Idp_save[(Vgp_save<1.2)&(Vgp_save>0.3)]
```

In [67]:

```
# fit in the valid range (you may want to go back and add the fitted line in the
fit = np.polyfit(Vg_clip,Ids_clip,1)
print('slope:',fit[0])
plt.rcParams.update({'font.size': 14})
plt.title('pFET ohmic region')
plt.plot(Vgp_save, Idp_save, '.k-')
plt.plot(Vgp_save, np.clip(Vgp_save*fit[0]+fit[1],a_min=0,a_max = 1),'.r--')
plt.xlabel('Vg(V)')
plt.ylabel('Ids(A)')
plt.legend(['measurements','fitting'])
plt.grid()
plt.show()
```

```
slope: -4.392172585087262e-07
```



(c) Determine V_{T0} and β for both devices by fitting your data to the expression derived in the prelab

In [68]:
```
# V_T0
v_t0p = -fit[1]/fit[0]
print('V_T0 =',v_t0p-1.8, 'V')
```

```
V_T0 = -0.5003495204460962 V
```

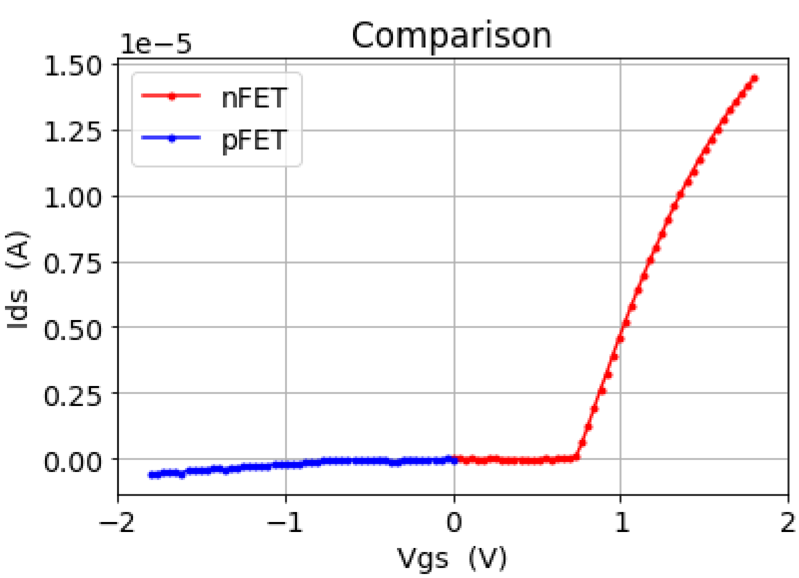
In [69]:
```
# beta
vds = -0.08
betap = fit[0]/vds
print('beta_p =',betap*1e6, 'uA/V/V')
```

```
beta_p = 5.4902157313590765 uA/V/V
```

5.3 Comparisons

- Include a single plot showing the curves for both devices.

In [79]:
```
# plot both Ids vs |Vgs|
import numpy as np
import matplotlib.pyplot as plt
Vgn_save, Isn_save = np.loadtxt('./data/Lab2_data_nFETVgIds.csv',delimiter=",")
plt.rcParams.update({'font.size': 14})
plt.plot(Vgn_save, Isn_save, 'r.-')
Vgp_save, Idp_save = np.loadtxt('./data/Lab2_data_pFETVgIds.csv',delimiter=",")
plt.rcParams.update({'font.size': 14})
plt.plot(-Vgn_save, -Idp_save, 'b.-')
plt.title('Comparison')
plt.xlabel('Vgs  (V)')
plt.ylabel('Ids  (A)')
plt.xlim([-2,2])
plt.legend(['nFET','pFET'])
plt.grid()
plt.show()
```



- What is the ratio between β for the 2 devices? Does it make sense?

In [58]:
```
ratio = betan/betap
print('The ratio between beta of nFET and pFET is:', ratio)
```

```
The ratio between beta of nFET and pFET is: 38.80561956497963
```

It does not make sense, the only difference between nFET and pFET according to the calculation for β should only be the differences between hole mobility and electron mobility, however, there difference is just around 3, while the difference here is too large.

It does not make sense, the only difference between nFET and pFET according to the calculation for β should only be the differences between hole mobility and electron mobility, however, there difference is just around 3, while the difference here is too large.* Is the relationship between I_{ds} and $V_{gs}-V_T$ really linear? What is likely the cause of any discrepancy?

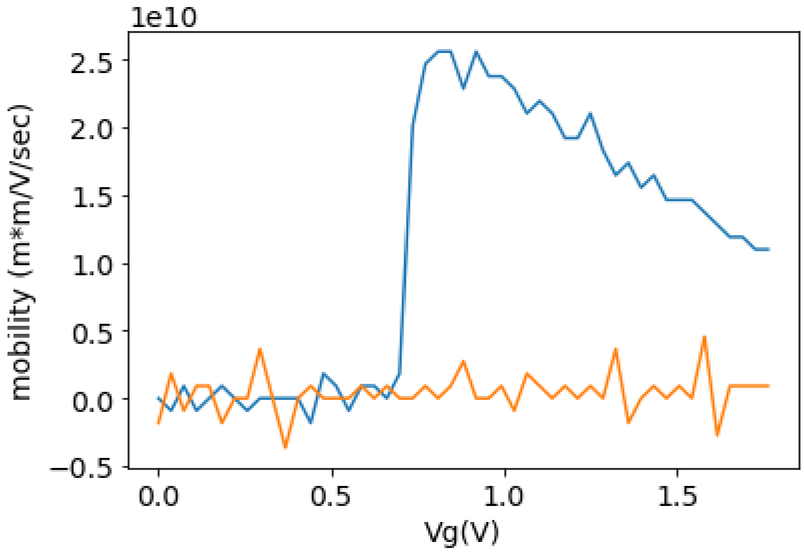
In the measurements for both case, the relationship is not really linear especially for nFET. And the possible cause is that when $V_{gs}-V_{T}$ increses, the mobility of electron μ_{n} decreases.

5.4 Effective surface mobility (optional)

Hint: Use the V_{T0} you obtained in the last experiments but assume β changes with V_{gs} (thus μ_n and μ_p changes). **No need to measure again.**

In [104…
```
# plot mu vs Vgs for both devices in the same figure
# plot both Ids vs |Vgs|
import numpy as np
import matplotlib.pyplot as plt
Vgn_save, Isn_save = np.loadtxt('./data/Lab2_data_nFETVgIds.csv',delimiter=",")
Vgp_save, Idp_save = np.loadtxt('./data/Lab2_data_pFETVgIds.csv',delimiter=",")
slopes_n = np.diff(Isn_save)/np.diff(Vgn_save)
slopes_p = np.diff(Idp_save)/np.diff(Vgp_save)
vds_n = 0.08
vds_p = -0.08
x = np.arange(len(slopes_n))/len(slopes_n)*1.8
mu_n = slopes_n/vds_n/9.09/1e-15
plt.plot(x,mu_n)
mu_p = slopes_p/vds_p/9.09/1e-15
plt.plot(x,mu_p)
plt.xlabel('Vg(V)')
plt.ylabel('mobility (m*m/V/sec)')
```

Out[104…
```
Text(0, 0.5, 'mobility (m*m/V/sec)')
```



- Why does the mobility peak and then decay instead of remaining constant?

With increasing electric field, the mobile charge carriers start to collide which reduces the increase of velocity, and since mobility is defined as $\mu = \frac{dvelocity}{dVoltage}$, the mobility drops.

- What is the ratio between the peak mobilities for electrons and holes?

around 6

- How different are these values from the bulk mobilities for electrons (1350 ${\rm cm^2/V/s}$) and holes (480 ${\rm cm^2/V/s}$)?

with the difference of 10^{11}, and the reason may be because the capacitance used here is per square micron.

6 Drain Current in the saturation region

In this experiment you will characterize the *quadratic* dependence of the current on the gate voltage in the saturation region.

6.1 N-FET

(a) Configure the chip following Section 4.2 if you haven't

(b) Measure I_{ds} as a function of V_g in saturation region

- What will be the fixed value for source and drain voltages?

In [11]:

```
## configure NMOS by AER event
# Configure NFET, set the input voltage demultiplexer by AER event.
# Note selectlines we should choose for the NFET
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine2, \
```

```
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]
p.send_coach_events(events)
```

In [12]:
```
# set source voltage
vs = 0.0
Vs_monitor = p.set_voltage(pyplane.DacChannel.GO20,vs)
print("The source voltage is set to {} V".format(Vs_monitor))
```

```
The source voltage is set to 0.0 V
```

In [13]:
```
# set drain voltage     #######1.8
vd = 1.8
Vd_monitor =p.set_voltage(pyplane.DacChannel.GO22,vd)
print("The drain voltage is set to {} V".format(Vd_monitor))
```

```
The drain voltage is set to 1.7982406616210938 V
```

- Data aquisition

In [14]:
```
# sweep gate voltage
# sweep gate voltage
import time
import numpy as np

# Get the leakage current, Read Ids=Ids0 at Vg = 0
p.set_voltage(pyplane.DacChannel.AIN0,0.0)
time.sleep(0.5) # wait 0.5 second for it to settle
Is0_n =  p.read_current(pyplane.AdcChannel.GO20_N)
print("Offset Is0_n: {} A".format(Is0_n))

number = 50
Vg_sets = np.linspace(0,1.8,num = number).reshape(-1,1)
Is_sets = np.zeros(number).reshape(-1,1)

for n in range(number):
    # set gate voltage
    p.set_voltage(pyplane.DacChannel.AIN0,Vg_sets[n])
    vg_set = p.get_set_voltage(pyplane.DacChannel.AIN0)
    print("The gate voltage is set to {} V".format(vg_set))   ## print the gate
    time.sleep(0.05)  # wait for it to settle

    # read I_{ds}
    Is_set = p.read_current(pyplane.AdcChannel.GO20_N)

    print("The measured source current is {} A".format(Is_set))  ## print the ra

    # substract leakage current
    Is_sets[n] = Is_set - Is0_n
```

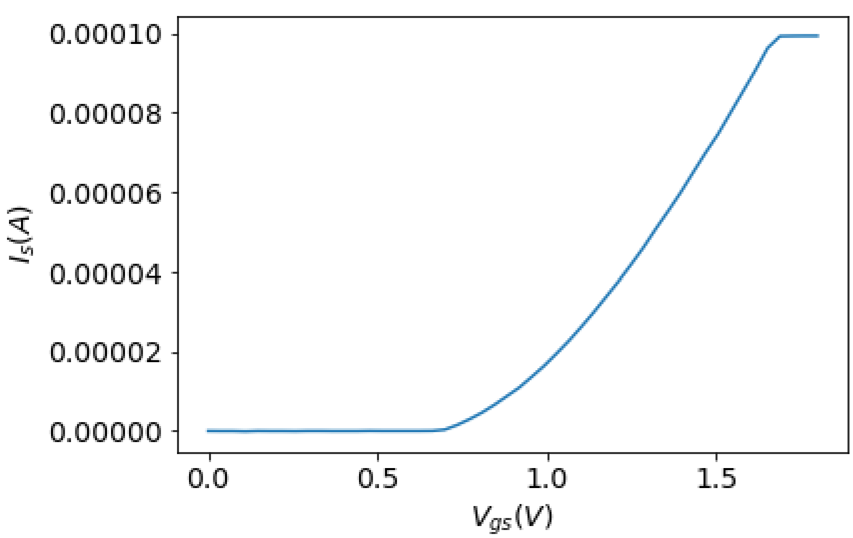
```
Offset Is0_n: 6.103515488575795e-07 A
The gate voltage is set to 0.0 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.03519061952829361 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.07214076817035675 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.10909092426300049 V
```

```
The measured source current is 5.37109372089617e-07 A
The gate voltage is set to 0.14604108035564423 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.18299122154712677 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.2199413776397705 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.25689151883125305 V
The measured source current is 5.859375278305379e-07 A
The gate voltage is set to 0.293841689825058 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.329032301902771 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.36598244309425354 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.4029325842857361 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.439882755279541 V
The measured source current is 6.103515488575795e-07 A
The gate voltage is set to 0.47683289647102356 V
The measured source current is 6.591797045985004e-07 A
The gate voltage is set to 0.5137830376625061 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.550733208656311 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.587683379650116 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.6228739619255066 V
The measured source current is 6.347656267280399e-07 A
The gate voltage is set to 0.6598241329193115 V
The measured source current is 6.591797045985004e-07 A
The gate voltage is set to 0.6967743039131165 V
The measured source current is 9.521484116703505e-07 A
The gate voltage is set to 0.7337244153022766 V
The measured source current is 2.1240234673314262e-06 A
The gate voltage is set to 0.7706745862960815 V
The measured source current is 3.588867230064352e-06 A
The gate voltage is set to 0.8076247572898865 V
The measured source current is 5.2490236157609615e-06 A
The gate voltage is set to 0.8445748686790466 V
The measured source current is 7.202148481155746e-06 A
The gate voltage is set to 0.8815250396728516 V
The measured source current is 9.375000445288606e-06 A
The gate voltage is set to 0.9167156219482422 V
The measured source current is 1.1547851499926765e-05 A
The gate voltage is set to 0.9536657929420471 V
The measured source current is 1.423339836037485e-05 A
The gate voltage is set to 0.990615963935852 V
The measured source current is 1.699218773865141e-05 A
The gate voltage is set to 1.0275660753250122 V
The measured source current is 2.0092773411306553e-05 A
The gate voltage is set to 1.064516305923462 V
The measured source current is 2.3315429643844254e-05 A
The gate voltage is set to 1.101466417312622 V
The measured source current is 2.6831054128706455e-05 A
The gate voltage is set to 1.1384165287017822 V
The measured source current is 3.0493163649225608e-05 A
The gate voltage is set to 1.175366759300232 V
The measured source current is 3.4374999813735485e-05 A
The gate voltage is set to 1.2105573415756226 V
The measured source current is 3.825683597824536e-05 A
```

```
The gate voltage is set to 1.2475074529647827 V
The measured source current is 4.238281326252036e-05 A
The gate voltage is set to 1.2844576835632324 V
The measured source current is 4.6679688239237294e-05 A
The gate voltage is set to 1.3214077949523926 V
The measured source current is 5.139160202816129e-05 A
The gate voltage is set to 1.3583579063415527 V
The measured source current is 5.5883789173094556e-05 A
The gate voltage is set to 1.3953081369400024 V
The measured source current is 6.052245953469537e-05 A
The gate voltage is set to 1.4322582483291626 V
The measured source current is 6.555175787070766e-05 A
The gate voltage is set to 1.4692083597183228 V
The measured source current is 7.050781277939677e-05 A
The gate voltage is set to 1.504399061203003 V
The measured source current is 7.517090125475079e-05 A
The gate voltage is set to 1.541349172592163 V
The measured source current is 8.051758049987257e-05 A
The gate voltage is set to 1.5782992839813232 V
The measured source current is 8.586425974499434e-05 A
The gate voltage is set to 1.615249514579773 V
The measured source current is 9.118652087636292e-05 A
The gate voltage is set to 1.652199625968933 V
The measured source current is 9.687500278232619e-05 A
The gate voltage is set to 1.6891497373580933 V
The measured source current is 9.985351789509878e-05 A
The gate voltage is set to 1.726099967956543 V
The measured source current is 9.990234684664756e-05 A
The gate voltage is set to 1.7630500793457031 V
The measured source current is 9.992675768444315e-05 A
The gate voltage is set to 1.7982406616210938 V
The measured source current is 9.992675768444315e-05 A
```

In [15]:

```python
# plot
# plot
import matplotlib.pyplot as plt
plt.plot(Vg_sets,Is_sets)
plt.xlabel(r'$V_{gs} (V)$')
plt.ylabel(r'$I_{s} (A)$')
plt.show()
```

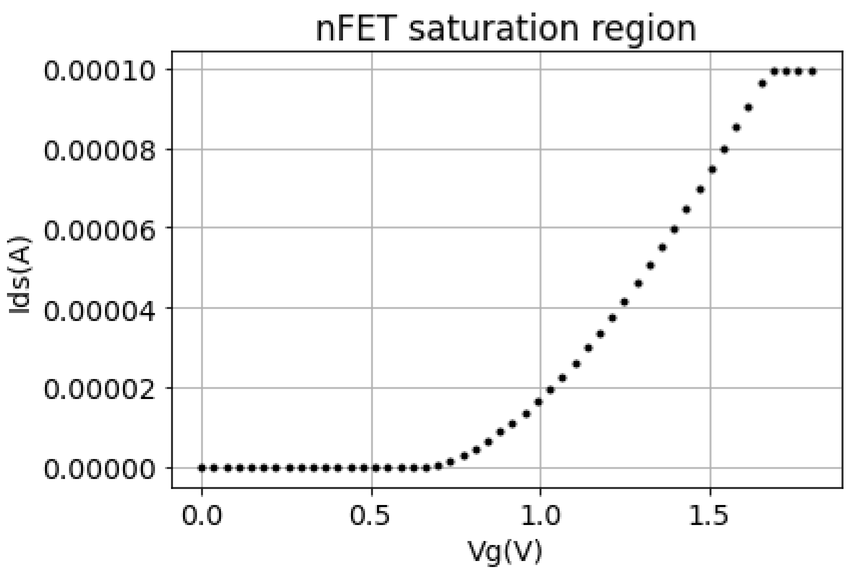


In [ ]:

```
# if the data looks nice, save it!
Lab2_data_6_nFETVgIds_Omic = [Vg_sets.reshape(number),Is_sets.reshape(number)]
np.savetxt('./data/Lab2_data_6_nFETVgIds.csv', Lab2_data_6_nFETVgIds_Omic, delim
```

In [107…

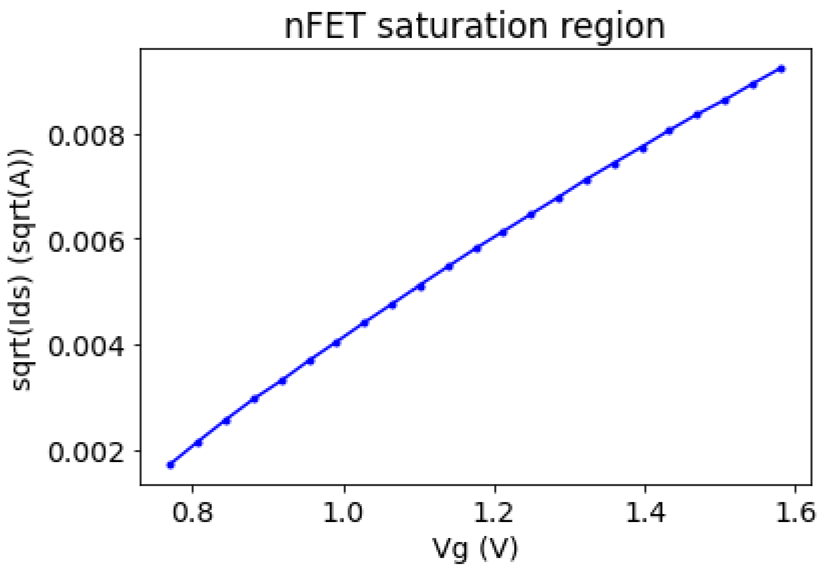
```
# example :
# Lab2_data_nFETVgIds_Omic = [Vg_n,Is_n]
# save to csv file
# np.savetxt('./data/Lab2_data_nFETVgIds.csv', Lab2_data_nFETVgIds_Omic, delimit
# Load data you saved and plot, to check if the data is saved correctly or not
Vgn_save, Isn_save = np.loadtxt('./data/Lab2_data_6_nFETVgIds.csv',delimiter=",'
plt.rcParams.update({'font.size': 14})
plt.plot(Vgn_save, Isn_save, '.k')
plt.xlabel('Vg(V)')
plt.ylabel('Ids(A)')
plt.title('nFET saturation region')
plt.grid()
plt.show()
```



In [112…

```
# extract the valid range and plot sqrt(Ids) vs Vgs
I_clip = Isn_save[(Vgn_save>0.75)&(Vgn_save<1.6)]
V_clip = Vgn_save[(Vgn_save>0.75)&(Vgn_save<1.6)]
plt.plot(V_clip, np.sqrt(I_clip),'.b-')
plt.xlabel('Vg (V)')
plt.ylabel('sqrt(Ids) (sqrt(A))')
plt.title('nFET saturation region')
```

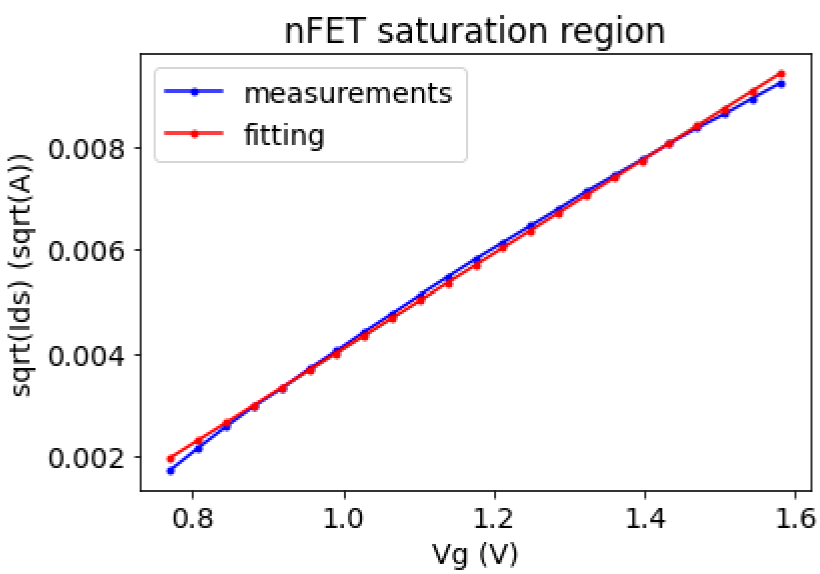
Out[112… Text(0.5, 1.0, 'nFET saturation region')



In [115…

```
# fit in the valid range (you may want to go back and add the fitted line in the
I_clip = Isn_save[(Vgn_save>0.75)&(Vgn_save<1.6)]
V_clip = Vgn_save[(Vgn_save>0.75)&(Vgn_save<1.6)]
plt.plot(V_clip, np.sqrt(I_clip),'.b-')
fit = np.polyfit(V_clip,np.sqrt(I_clip),1)
plt.plot(V_clip,V_clip*fit[0]+fit[1],'.r-')
plt.xlabel('Vg (V)')
plt.ylabel('sqrt(Ids) (sqrt(A))')
plt.title('nFET saturation region')
plt.legend(['measurements','fitting'])
print('slope =:',fit[0])
```

```
slope =: 0.00922661160104714
```



(c) Determine V_{T0} and β for both devices by fitting your data to the expression derived in the prelab

In [116…

```
# V_T0
print('V_T0 = ', -fit[1]/fit[0], 'V' )
```

```
V_T0 =  0.5584695102935967
```

In [118…

```
# beta
betan = 2*fit[0]**2
print('beta = ',betan*1e6, 'uA/V/V')
```

```
beta =  170.26072327315532 uA/V/V
```

6.2 P-FET

(a) Configure the chip following Section 4.3 if you haven't

(b) Measure I_{ds} as a function of V_g in ohmic region

- What will be the fixed value for bulk, source and drain voltages?

In [6]:

```
# Configure PFET, set the input voltage demultiplexer by AER event.
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine1, \
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]

p.send_coach_events(events)
```

In [8]:

```
# set bulk voltage
vb = 1.8
p.set_voltage(pyplane.DacChannel.AIN1,vb)
time.sleep(0.05)  # wait for it to settle

# set source voltage
vs = 1.8
p.set_voltage(pyplane.DacChannel.GO23,vs)
time.sleep(0.05)  # wait for it to settle

# set drain voltage
vd = 0.0
p.set_voltage(pyplane.DacChannel.GO21,vd)
time.sleep(0.05)  # wait for it to settle

# set gate voltage
vg = 1.8
p.set_voltage(pyplane.DacChannel.AIN0,vg)
time.sleep(0.05)  # wait for it to settle


# Print I_ds for checking
I_ds =  p.read_current(pyplane.AdcChannel.GO21_N)
print("Offset I_ds: {} A".format(I_ds))
```

```
Offset I_ds: 4.1503906800244295e-07 A
```

- Data aquisition

In [10]:

```
# sweep gate voltage
# sweep gate voltage
import time
import numpy as np
```

```
# Get the leakage current, Read Ids=Ids0 at Vg = 1.8
p.set_voltage(pyplane.DacChannel.AIN0,1.8)
time.sleep(0.5) # wait 0.5 second for it to settle
Id0_p =  p.read_current(pyplane.AdcChannel.GO21_N)
print("Offset Id0_p: {} A".format(Id0_p))


number = 50
Vg_sets = np.linspace(1.8,0.0,num = number).reshape(-1,1)
Id_sets = np.zeros(number).reshape(-1,1)
Is_sets = np.zeros(number).reshape(-1,1)
for n in range(number):
    # set gate voltage
    p.set_voltage(pyplane.DacChannel.AIN0,Vg_sets[n])
    vg_set = p.get_set_voltage(pyplane.DacChannel.AIN0)
    print("The gate voltage is set to {} V".format(vg_set))   ## print the gate
    time.sleep(0.05)  # wait for it to settle

    # read I_{ds}
    Id_set = p.read_current(pyplane.AdcChannel.GO21_N)
    print("The measured drain current is {} A".format(Id_set))  ## print the raw

    # substract leakage current
    Id_sets[n] = Id_set - Id0_p
```

```
Offset Id0_p: 4.1503906800244295e-07 A
The gate voltage is set to 1.7982406616210938 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.7630500793457031 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.726099967956543 V
The measured drain current is 4.39453117451194e-07 A
The gate voltage is set to 1.6891497373580933 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.652199625968933 V
The measured drain current is 5.126952942191565e-07 A
The gate voltage is set to 1.615249514579773 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.5782992839813232 V
The measured drain current is 4.39453117451194e-07 A
The gate voltage is set to 1.541349172592163 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.504399061203003 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.4692083597183228 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.4322582483291626 V
The measured drain current is 4.39453117451194e-07 A
The gate voltage is set to 1.3953081369400024 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.3583579063415527 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.3214077949523926 V
The measured drain current is 4.39453117451194e-07 A
The gate voltage is set to 1.2844576835632324 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.2475074529647827 V
The measured drain current is 4.1503906800244295e-07 A
The gate voltage is set to 1.2105573415756226 V
The measured drain current is 4.39453117451194e-07 A
```

```
The gate voltage is set to 1.175366759300232 V
The measured drain current is 4.39453117451194e-07 A
The gate voltage is set to 1.1384165287017822 V
The measured drain current is 4.39453117451194e-07 A
The gate voltage is set to 1.101466417312622 V
The measured drain current is 5.615234499600774e-07 A
The gate voltage is set to 1.064516305923462 V
The measured drain current is 8.78906234902388e-07 A
The gate voltage is set to 1.0275660753250122 V
The measured drain current is 1.2939452744831215e-06 A
The gate voltage is set to 0.990615963935852 V
The measured drain current is 1.7089844277506927e-06 A
The gate voltage is set to 0.9536657929420471 V
The measured drain current is 2.197265530412551e-06 A
The gate voltage is set to 0.9167156219482422 V
The measured drain current is 2.7832031719299266e-06 A
The gate voltage is set to 0.8815250396728516 V
The measured drain current is 3.344726565046585e-06 A
The gate voltage is set to 0.8445748686790466 V
The measured drain current is 4.0039062696450856e-06 A
The gate voltage is set to 0.8076247572898865 V
The measured drain current is 4.785156306752469e-06 A
The gate voltage is set to 0.7706745862960815 V
The measured drain current is 5.468749805004336e-06 A
The gate voltage is set to 0.7337244153022766 V
The measured drain current is 6.24999984211172e-06 A
The gate voltage is set to 0.6967743039131165 V
The measured drain current is 7.080077921273187e-06 A
The gate voltage is set to 0.6598241329193115 V
The measured drain current is 7.958984497236088e-06 A
The gate voltage is set to 0.6228739619255066 V
The measured drain current is 8.837891073198989e-06 A
The gate voltage is set to 0.587683379650116 V
The measured drain current is 9.765624781721272e-06 A
The gate voltage is set to 0.550733208656311 V
The measured drain current is 1.0791015483846422e-05 A
The gate voltage is set to 0.5137830376625061 V
The measured drain current is 1.1840820661745965e-05 A
The gate voltage is set to 0.47683289647102356 V
The measured drain current is 1.293945297220489e-05 A
The gate voltage is set to 0.439882755279541 V
The measured drain current is 1.411132780049229e-05 A
The gate voltage is set to 0.4029325842857361 V
The measured drain current is 1.52587890625e-05 A
The gate voltage is set to 0.36598244309425354 V
The measured drain current is 1.657714892644435e-05 A
The gate voltage is set to 0.329032301902771 V
The measured drain current is 1.7700194803182967e-05 A
The gate voltage is set to 0.293841689825058 V
The measured drain current is 1.889648410724476e-05 A
The gate voltage is set to 0.25689151883125305 V
The measured drain current is 2.0288085579522885e-05 A
The gate voltage is set to 0.2199413776397705 V
The measured drain current is 2.1533203835133463e-05 A
The gate voltage is set to 0.18299122154712677 V
The measured drain current is 2.2949217964196578e-05 A
The gate voltage is set to 0.14604108035564423 V
The measured drain current is 2.4365233912249096e-05 A
The gate voltage is set to 0.10909092426300049 V
The measured drain current is 2.5805664336076006e-05 A
The gate voltage is set to 0.07214076817035675 V
```

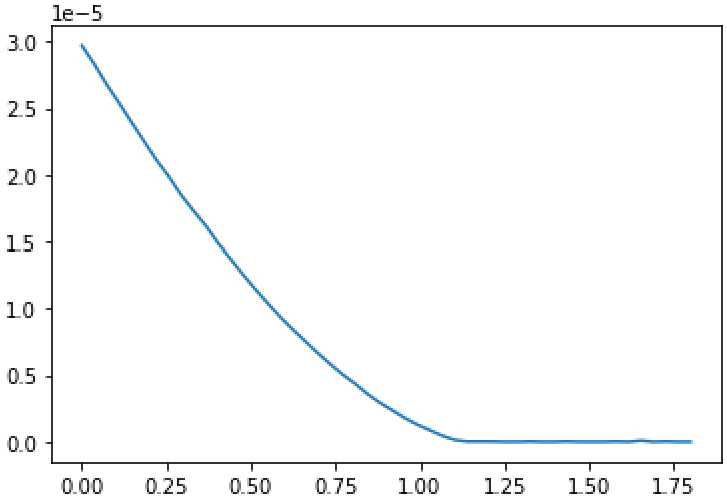
```
The measured drain current is 2.7197265808354132e-05 A
The gate voltage is set to 0.03519061952829361 V
The measured drain current is 2.873535231628921e-05 A
The gate voltage is set to 0.0 V
The measured drain current is 3.0126953788567334e-05 A
```

In [11]:

```
# plot
# plot
import matplotlib.pyplot as plt
plt.plot(Vg_sets,Id_sets)
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x7f5745eb37f0>]
```

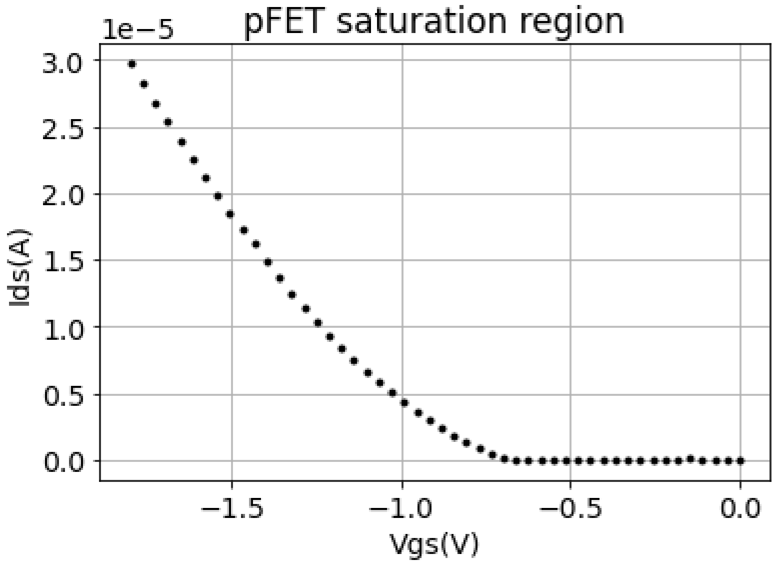


In [12]:

```
# if the data looks nice, save it!
Lab2_data_6_pFETVgIds_Omic = [Vg_sets.reshape(number),Id_sets.reshape(number)]
np.savetxt('./data/Lab2_data_6_pFETVgIds.csv', Lab2_data_6_pFETVgIds_Omic, delim
```

In [121…

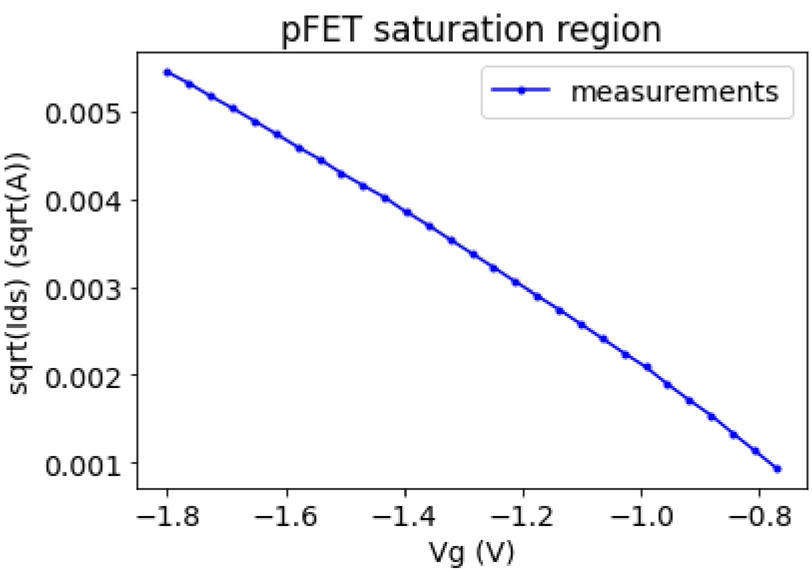
```
Vgp_save, Isp_save = np.loadtxt('./data/Lab2_data_6_pFETVgIds.csv',delimiter=",'
plt.rcParams.update({'font.size': 14})
plt.plot(Vgp_save-1.8, Isp_save, '.k')
plt.xlabel('Vgs(V)')
plt.ylabel('Ids(A)')
plt.title('pFET saturation region')
plt.grid()
plt.show()
```



In [135…

```
# extrat the valid range and plot sqrt(Ids) vs Vgs
V_clip = Vgp_save[Vgp_save<-0.75+1.8]
I_clip = Isp_save[Vgp_save<-0.75+1.8]
plt.plot(V_clip-1.8, np.sqrt(I_clip),'.b-')
plt.xlabel('Vg (V)')
plt.ylabel('sqrt(Ids) (sqrt(A))')
plt.title('pFET saturation region')
plt.legend(['measurements','fitting'])
print('slope =:',fit[0])
```

slope =: -0.004343458044805485

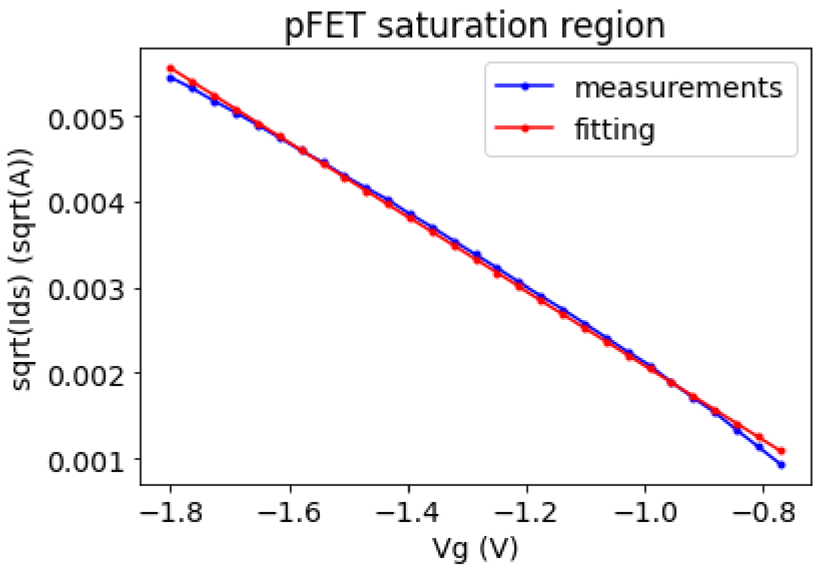


In [136…

```
# fit in the valid range (you may want to go back and add the fitted line in the
# extrat the valid range and plot sqrt(Ids) vs Vgs
V_clip = Vgp_save[Vgp_save<-0.75+1.8]
I_clip = Isp_save[Vgp_save<-0.75+1.8]
plt.plot(V_clip-1.8, np.sqrt(I_clip),'.b-')
fit = np.polyfit(V_clip,np.sqrt(I_clip),1)
plt.plot(V_clip-1.8,V_clip*fit[0]+fit[1],'.r-')
plt.xlabel('Vg (V)')
plt.ylabel('sqrt(Ids) (sqrt(A))')
plt.title('pFET saturation region')
```

```
plt.legend(['measurements','fitting'])
print('slope =:',fit[0])
```

```
slope =: -0.004343458044805485
```



(c) Determine V_{T0} and β for both devices by fitting your data to the expression derived in the prelab

In [138…

```
# V_T0
print('V_T0 = ', -fit[1]/fit[0]-1.8, 'V' )
```

```
V_T0 =  -0.5200040826262322 V
```

In [140…

```
# beta
betan = 2*fit[0]**2
print('beta = ',betan*1e6, 'uA/V/V')
```

```
beta =  37.73125557397098 uA/V/V
```

6.3 Comparisons

- Are the measurements of V_{T0} and β from the saturation measurement consistent with the values obtained in the ohmic region?

The measurements of V_{T0} is consistent while β is not.

- Which is a better approximation, the linear one or the quadratic?

The quadratic one, since the measurement noise is smaller and β makes more sense.

7 Early effect

This experiment studies how Early voltage scales with transistor current; in particular, how valid are the simple assumptions about channel length modulation?

You only need to do N-FET

(a) Measure \$I_{ds}\$ vs \$V_{ds}\$ for different \$V_{gs}\$

In [7]:

```
### AER to configure NMOS
## configure NMOS by AER event
# Configure NFET, set the input voltage demultiplexer by AER event.
# Note selectlines we should choose for the NFET
events = [pyplane.Coach.generate_aerc_event( \
    pyplane.Coach.CurrentOutputSelect.SelectLine5, \
    pyplane.Coach.VoltageOutputSelect.NoneSelected, \
    pyplane.Coach.VoltageInputSelect.SelectLine2, \
    pyplane.Coach.SynapseSelect.NoneSelected, 0)]
p.send_coach_events(events)
```

In [8]:

```
# set source voltage
vs = 0.0
Vs_monitor = p.set_voltage(pyplane.DacChannel.GO20,vs)
print("The source voltage is set to {} V".format(Vs_monitor))
```

```
The source voltage is set to 0.0 V
```

In [9]:

```
# Measurement. You may need two 'for' loops (one nested loop) to sweep Vgs and V
import time
import numpy as np

# Get the leakage current, Read Ids=Ids0 at Vg = 0
p.set_voltage(pyplane.DacChannel.AIN0,0.0)
time.sleep(0.5) # wait 0.5 second for it to settle
Is0_n =  p.read_current(pyplane.AdcChannel.GO20_N)
print("Offset Is0_n: {} A".format(Is0_n))

num_Vg = 10
num_Vd = 50
Vg_sets = np.linspace(0.6,1.8,num = num_Vg).reshape(-1,1)
Vd_sets = np.linspace(0,1.8,num = num_Vd).reshape(-1,1)
Is_sets = np.zeros((num_Vd,num_Vg))

for i in range(num_Vg):
    p.set_voltage(pyplane.DacChannel.AIN0,Vg_sets[i])
    time.sleep(0.05)  # wait for it to settle
    for j in range(num_Vd):
        # set gate voltage
        p.set_voltage(pyplane.DacChannel.GO22,Vd_sets[j])
        time.sleep(0.05)  # wait for it to settle

        # read I_{ds}
        Is_set = p.read_current(pyplane.AdcChannel.GO20_N)
        # substract leakage current
        Is_sets[j,i] = Is_set - Is0_n
```

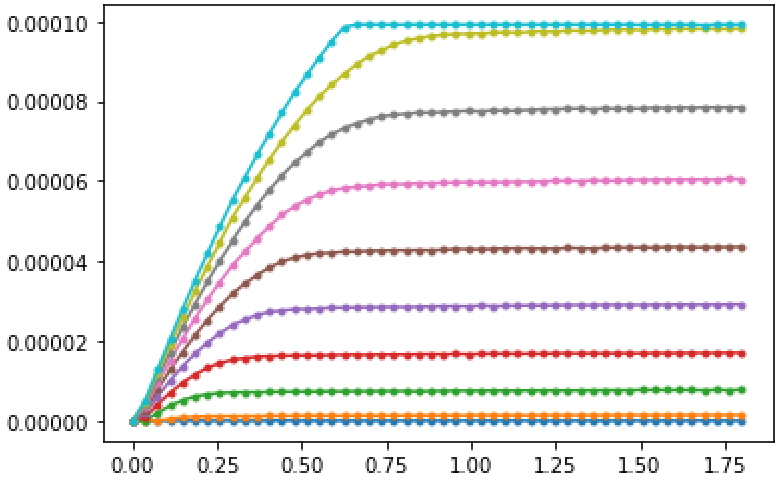
```
Offset Is0_n: 6.591797045985004e-07 A
```

- Include a single plot showing all data on a semilogy plot.

In [10]:

```
# plot
import matplotlib.pyplot as plt
```

```
for i in range(num_Vg):
    plt.plot(Vd_sets.reshape(-1,1),Is_sets[:,i].reshape(-1,1),'.-')
```



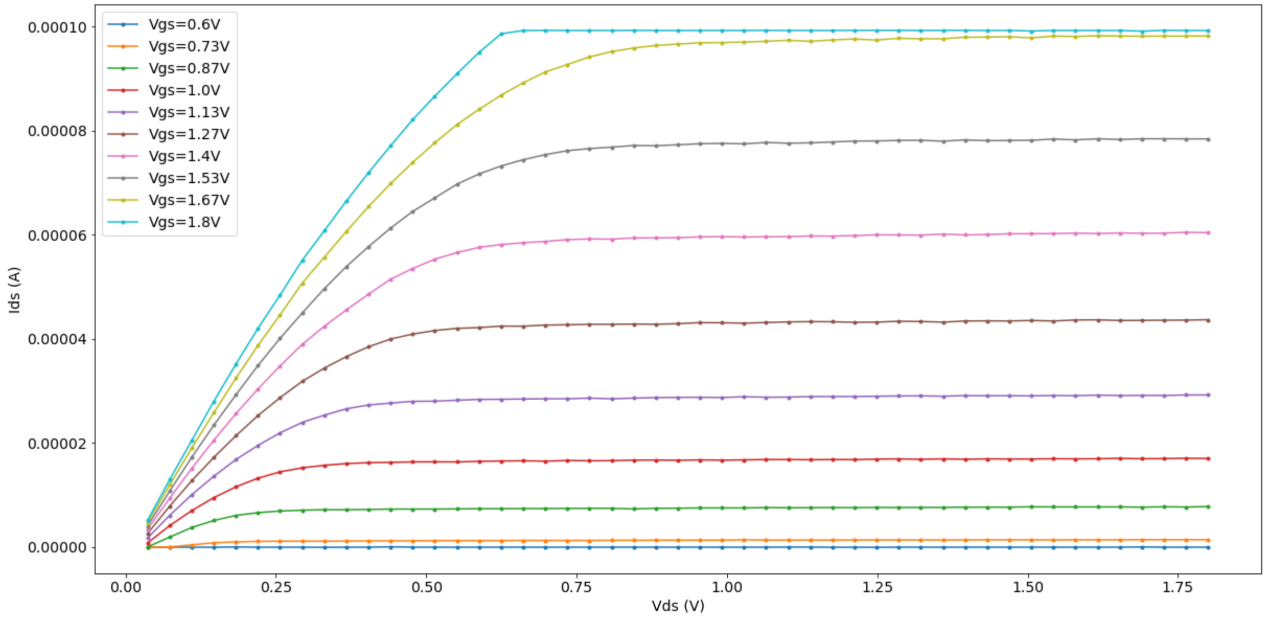
In [11]:

```
# if the data looks nice, save it!
import pandas as pd
I_data = np.zeros((num_Vd,num_Vg+1))
I_data[:,0] = Vd_sets.reshape(num_Vd)
for i in range(num_Vg):
    I_data[:,i+1] = Is_sets[:,i]
pd.DataFrame(I_data).to_csv("data_7_earlyeffect.csv",header=None, index=None)
```

In [160…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv ('data_7_earlyeffect.csv')
arr = pd.DataFrame(df).to_numpy()
print(arr.shape)
Vds = arr[:,0]
legends = []
for i in range(10):
    plt.plot(Vds,arr[:,i+1].reshape(-1,1),'.-')
    legends.append('Vgs='+str(round(0.6+i*1.2/9,2))+'V')
plt.xlabel('Vds (V)')
plt.ylabel('Ids (A)')
plt.legend(legends)
plt.rcParams["figure.figsize"] = [20,10]
```

```
(49, 11)
```



- Can you see how the saturation voltage increases with the gate overdrive V_G-V_T in strong inversion?

The saturation voltage increases first quadratically, and then it saturates.

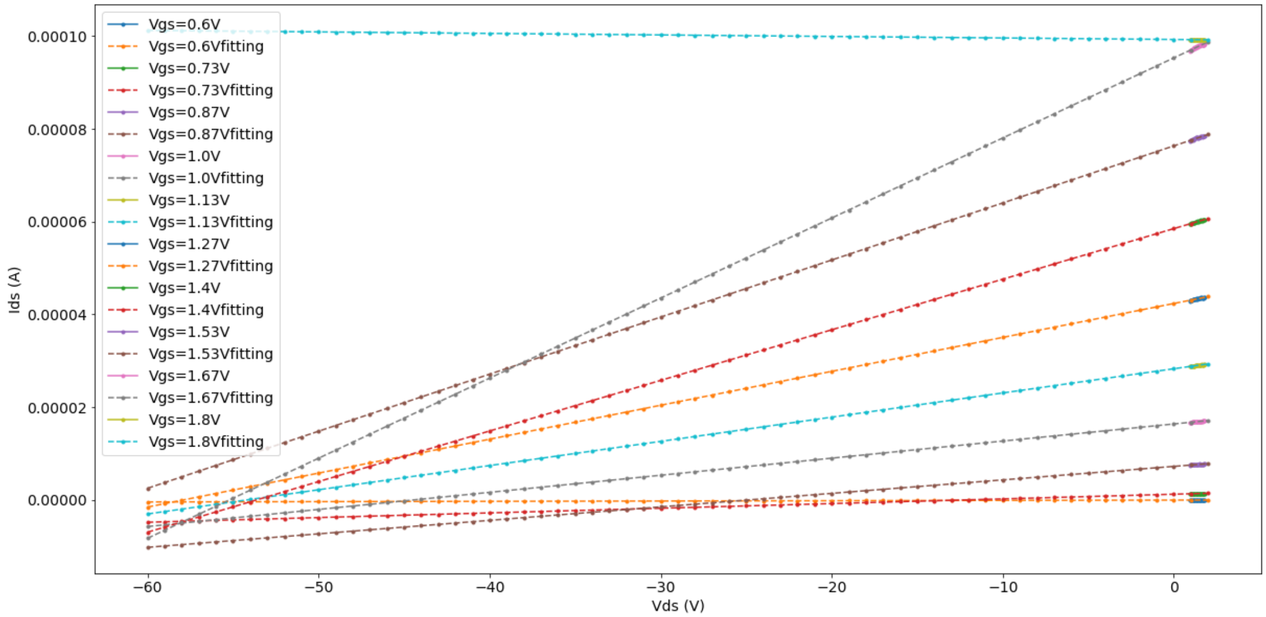
(b) Compute the Early voltage

- Fit a line to the ”flat” part of each curve. Select a range of drain voltages to fit the line and use the same range for each curve, because the Early effect is actually curved in reality, and what you are actually seeing is the start of Drain Induced Barrier Lowering (DIBL) or impact ionization.

In [167…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv ('data_7_earlyeffect.csv')
arr = pd.DataFrame(df).to_numpy()
print(arr.shape)
Vds = arr[:,0]
legends = []
Vds_range = np.linspace(-60,2,num = 63)
for i in range(10):
    plt.plot(Vds[Vds>0.9],arr[:,i+1][Vds>0.9],'.-')
    fit = np.polyfit(Vds[Vds>0.9],arr[:,i+1][Vds>0.9],1)
    legends.append('Vgs='+str(round(0.6+i*1.2/9,2))+'V')
    plt.plot(Vds_range, Vds_range*fit[0]+fit[1],'.--')
    legends.append('Vgs='+str(round(0.6+i*1.2/9,2))+'V'+'fitting')
plt.xlabel('Vds (V)')
plt.ylabel('Ids (A)')
plt.legend(legends)
plt.rcParams["figure.figsize"] = [20,10]
```

```
(49, 11)
```



- Plot the Early voltage vs drain current on a semilogx scale.

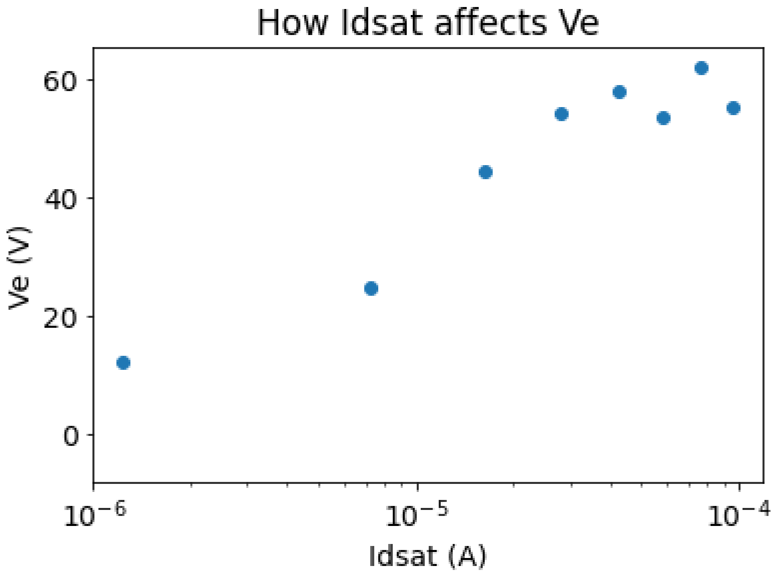
In [178…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv ('data_7_earlyeffect.csv')
arr = pd.DataFrame(df).to_numpy()
print(arr.shape)
Vds = arr[:,0]
Ve = np.zeros(9)
Idsat = np.zeros(9)
Vds_range = np.linspace(-60,2,num = 63)
for i in range(9):
    fit = np.polyfit(Vds[Vds>0.9],arr[:,i+1][Vds>0.9],1)
    Idsat[i] = fit[1]
    Ve[i] = -fit[1]/fit[0]
plt.semilogx(Idsat,-Ve,'o')
plt.rcParams["figure.figsize"] = [6,4]
plt.xlabel('Idsat (A)')
plt.ylabel('Ve (V)')
plt.title('How Idsat affects Ve')
```

```
(49, 11)
```

Out[178…

```
Text(0.5, 1.0, 'How Idsat affects Ve')
```



- Comment on your results: How constant is the Early voltage with drain current? Speculate on the reasons for your observations.

The early voltage first increases with drain saturation current and then decreases.

Because of 2nd order drain effects, the real curve is not flat.

8 Congratulations

If you did everything in this lab, you have done a lot! This is probably the most difficult but also one of the most important labs, because practical and intuitive knowledge of transistor characteristics is crucial in understanding and synthesizing new circuits.

9 What we expect

How transistors work above threshold.

What is the linear or triode region and what is the saturation region?

How does the linear region depend on gate and threshold voltage?

What is the *overdrive*?

What is the specific current?

How the Early effect comes about?

Typical values for Early voltage.

How to sketch graphs of transistor current vs gate voltage and drain-source voltage.

How above-threshold transistors go into saturation and why the saturation voltage is equal to the gate overdrive. Can you write the above-threshold current equations?

How does above-threshold current depend on W/L, C_{ox}, and mobility μ?

How do transconductance and drain resistance combine to generate voltage gain? And what is the intrinsic voltage gain of a transistor?

What effect does velocity saturation have on transistor operation, specifically, how does it change the relation between saturation current and gate voltage? What is DIBL (drain induced barrier lowering) and II (impact ionization)?

What is the dominant source of mismatch?

How does transistor mismatch scale with transistor size?

What are typical values of transistor threshold voltage mismatch?