

# 2021\_01\_06\_EE538\_Lecture1\_W2021

January 6, 2021

## 1 EE 538: Analog Integrated Circuit Design

### 1.1 Winter 2021

### 1.2 Instructor: Jason Silver

### 1.3 EE 538 basics I

- Instructor
  - Jason Silver
  - Office hours TBD via Google poll
- Teaching assistant
  - Thushara Maria Xavier
  - Office hours TBD
- Web page: [EE 538 Winter 2021 \(Canvas\)](#)
  - Access assignments, grades, and solutions
- Slack
  - [EE 538 Winter 2021 Slack workspace](#)
  - Use Slack to ask questions about assignments and projects
  - Participation in online discussion benefits everyone!

### 1.4 EE 538 basics II

- There are no official prerequisites, but it will be helpful to have some familiarity with
  - Elementary circuit theory
    - \* KVL, KCL, Thevenin equivalent circuits, Laplace/Fourier transforms
  - Semiconductor device operation and circuit analysis
    - \* Diodes, FETs, BJTs
  - Basic linear systems
    - \* Frequency response, poles, zeros, Bode plots
  - Circuit simulation with some flavor of SPICE (Simulation Program with Integrated Circuit Emphasis)

### 1.5 About your instructor

- PhD from UW EE in 2015
  - Low power integrated circuit (IC) design for bioelectrical interfaces
    - \* EEG, EMG, neural recording
  - Focus on optimizing for power efficiency

- 13 years experience designing ICs and systems for academia and industry
  - Mixed-signal design for biomedical applications
- Current full-time role
  - Director of Hardware and Biosystems Engineering at Curi Bio
    - \* Formerly housed in UW CoMotion startup incubator (Fluke Hall)
    - \* Instrumentation for *in vitro* cell studies

## 1.6 Course breakdown

- Weekly Assignments (40%)
  - Typically assigned Saturday, submitted online the Sunday of the following week
- Design project (40%)
  - Analog IC design project using Cadence tools
  - Optimization for performance, power, cost
- Midterm exam (20%)
  - Single exam covering approximately half of the course material
  - “Take-home” format, submitted online

## 1.7 Course learning goals

- Develop deeper understanding of MOS transistor behavior relevant to analog (and some digital) design
- Develop intuition w.r.t. tradeoffs in analog circuits (speed, noise, power dissipation)
- Learn to bridge the gap between complex device models/behavior and “back-of-the-envelope” calculations
- Develop a systematic approach to circuit analysis and design

## 1.8 Course topics

- CMOS technology and device models
- Single-stage amplifiers
- Current mirrors, active loads
- Differential pairs
- Operational transconductance amplifiers (OTAs)
- Feedback, stability, and compensation

## 1.9 Software and CAD

- We will use Cadence for circuit simulation
  - Tutorial following today’s lecture
- Design, data analysis, and results plotting using Python/Jupyter Notebooks
  - Design scripts iterable and reusable
  - More flexible than Cadence native plotting functions
  - Lecture examples created using Python/Jupyter Notebooks

## 1.10 SPICE design methodology

- SPICE is a numerical simulation tool that enables you to evaluate circuit ideas
- General rule: Don’t simulate something you don’t already (mostly) understand

- SPICE is for verification only!
- Neither analytical nor simulation models provide a complete picture of reality
  - Understanding model limitations is crucial to building successful circuits and systems

## 1.11 JupyterHub

- Jupyter Hub enables execution of Python code without the need for installation/maintenance of packages, etc
- Lecture notes/slides will be made available in student directories several days prior to lecture
- [EE538 Jupyter Hub Server](#)
- Please log out of the server when you're not using it!

## 1.12 Python packages/modules

```
[2]: import matplotlib as mpl
from matplotlib import pyplot as plt
import numpy as np
from scipy import signal
#%%matplotlib notebook

mpl.rcParams['font.size'] = 12
mpl.rcParams['legend.fontsize'] = 'large'

def plot_xy(x, y, xlabel, ylabel):
    fig, ax = plt.subplots(figsize=(10.0, 7.5));
    ax.plot(x, y, 'b');
    ax.grid();
    ax.set_xlabel(xlabel);
    ax.set_ylabel(ylabel);

def plot_xy2(x1, y1, x1label, y1label, x2, y2, x2label, y2label):
    fig, ax = plt.subplots(2, figsize = (10.0, 7.5));
    ax[0].plot(x1, y1, 'b');
    ax[0].set_ylabel(y1label)
    ax[0].grid()

    ax[1].plot(x2, y2, 'b');
    ax[1].set_xlabel(x1label)
    ax[1].set_xlabel(x2label);
    ax[1].set_ylabel(y2label);
    ax[1].grid();

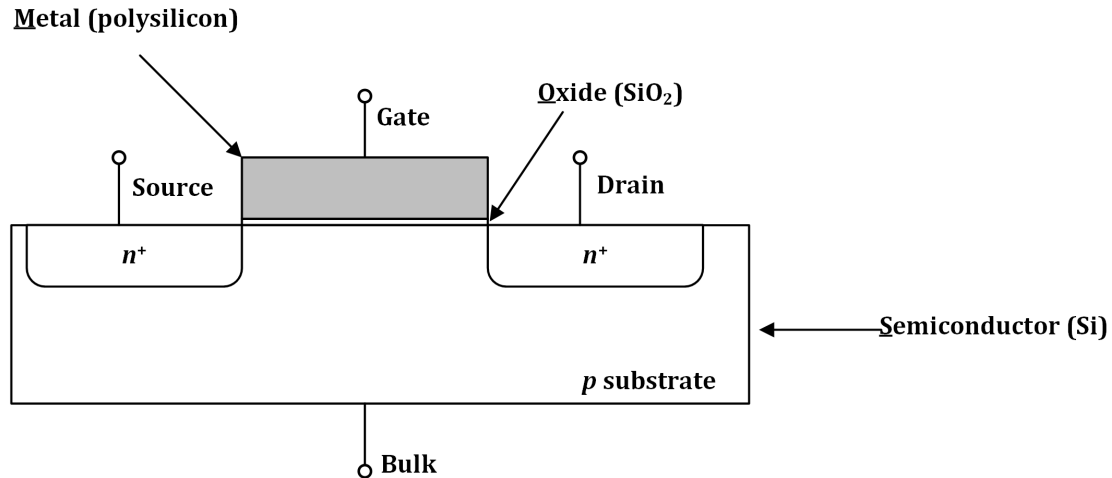
    fig.align_ylabels(ax[:])

def plot_xlogy(x, y, xlabel, ylabel):
    fig, ax = plt.subplots(figsize=(10.0, 7.5));
```

```
ax.semilogy(x, y, 'b');
ax.grid();
ax.set_xlabel(xlabel);
ax.set_ylabel(ylabel);
```

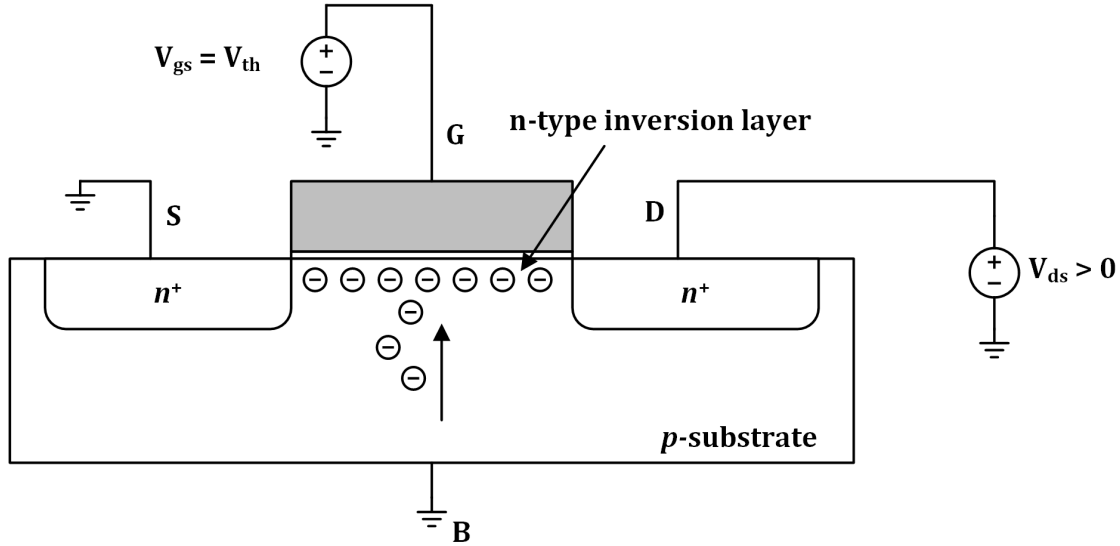
## 2 Lecture 1 - MOS Physics and Operation

### 2.1 MOS transistor



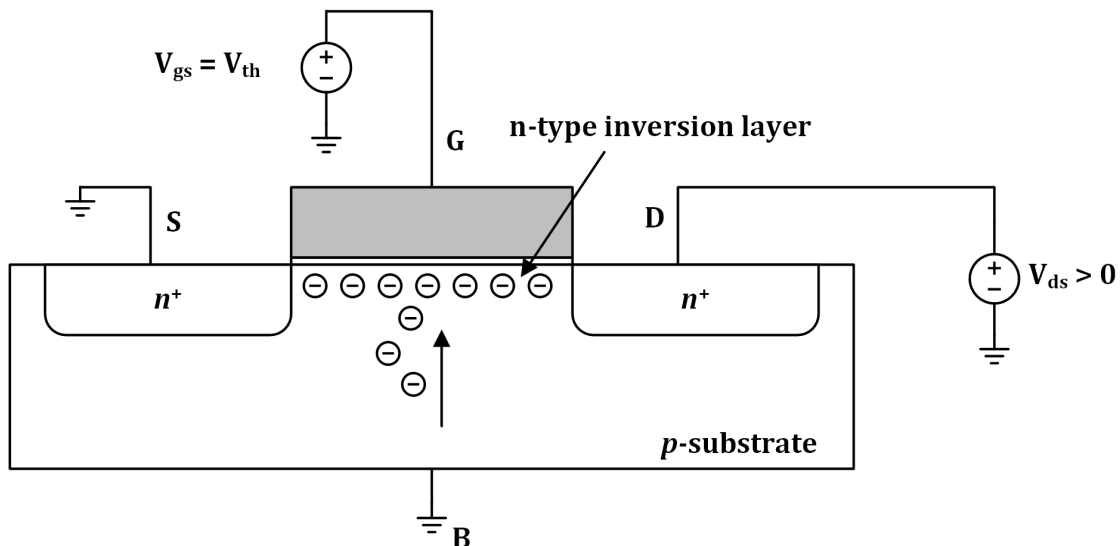
- MOSFET: **M**etal-**O**xide-**S**emiconductor **F**ield **E**ffect **T**ransistor
- CMOS: **C**omplementary **MOS** (NMOS and PMOS in a single process)
- n-type transistors (NMOS) consist of p-doped bulk, n-doped source/drain, polysilicon gate, and SiO<sub>2</sub> insulating layer
- p-type transistors (PMOS) have n-doped bulk, p-doped source/drain

## 2.2 NMOS operation



- In the absence of an applied gate-source voltage ( $V_{gs}$ ), the charge concentration under the gate is dominated by *majority* carriers (holes for an NMOS)
- An *inversion layer* begins to form as minority carriers are drawn from bulk to interface for  $V_{gs} > 0$
- Threshold voltage ( $V_{th}$ ) is defined as the  $V_{gs}$  value at which the minority carrier (electron) concentration equals that of the majority carriers (holes)

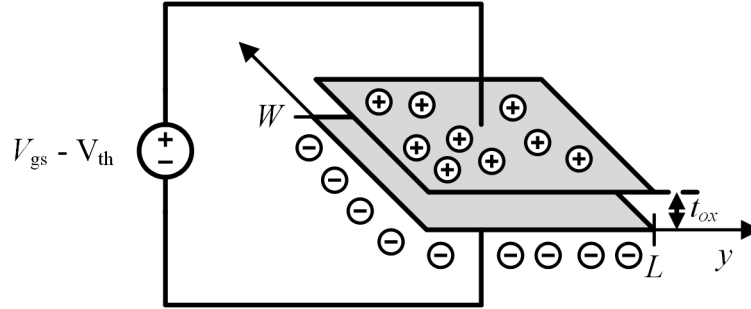
## 2.3 NMOS operation



- Current is controlled by the velocity of mobile charge in the channel
- Vertical electric field ( $E_{vert}$ ) controls charge density
- Carrier velocity proportional to lateral electric field ( $v = \mu E_{lat}$ )

- Our goal: Calculate drain current as a function of  $V_{gs}$  and  $V_{ds}$

## 2.4 Vertical electric field

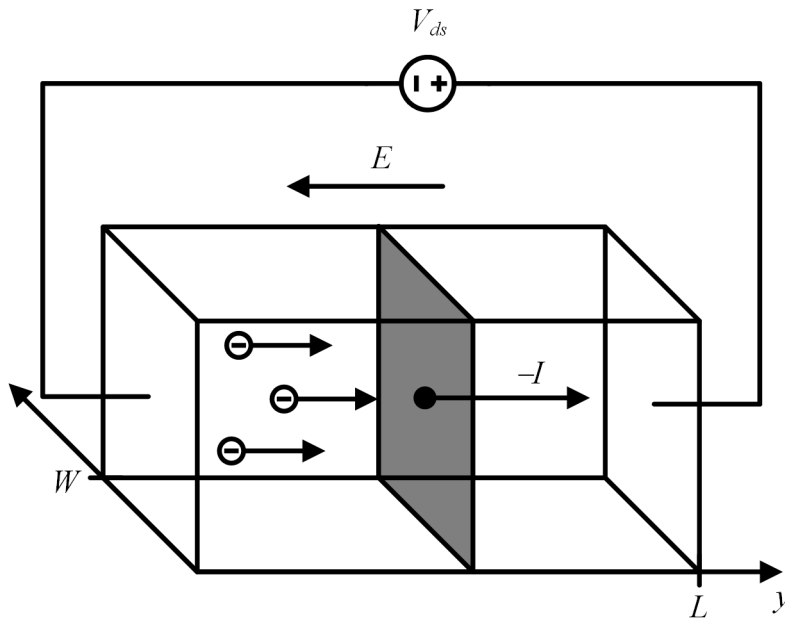


- $C_{ox}$ : Oxide capacitance, in  $F/m^2$
- Charge per unit area,  $Q(y)$ , depends on  $V_{gs} - V_{th}$ ,  $\epsilon_{ox}$ , and  $t_{ox}$ :

$$Q(y) = C_{ox}[V_{gs} - V_{th} - V(y)] = \frac{\epsilon_{ox}}{t_{ox}}[V_{gs} - V_{th} - V(y)] \quad (1)$$

- Vertical electric field controls channel charge density

## 2.5 Lateral electric field



- Current is controlled by charge density, mobility, and applied voltage:

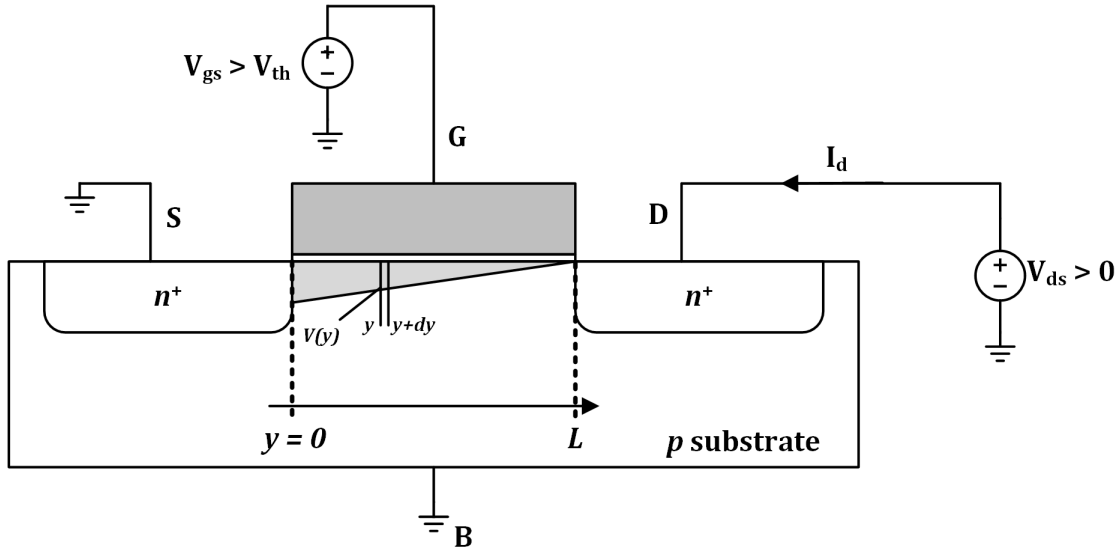
$$I = Q(y) \cdot v \quad (2)$$

$$= Q(y) \cdot \mu \cdot E_{lat} \quad (3)$$

$$= Q(y) \cdot \mu \cdot \frac{dV(y)}{dy} \quad (4)$$

- While the vertical electric field controls charge density, the lateral electric field controls charge velocity

## 2.6 First-order I-V characteristics



- Charge density and velocity:

$$Q_n(y) = C_{ox}[V_{gs} - V_{th} - V(y)] \quad (5)$$

$$v = \mu \cdot E_{lat} \quad (6)$$

- Resulting drain current:

$$I_d = Q_n \cdot v \cdot W \quad (7)$$

$$(8)$$

$$= C_{ox}[V_{gs} - V_{th} - V(y)] \cdot \mu \cdot E_{lat} \cdot W \quad (9)$$

$$(10)$$

## 2.7 MOS I-V derivation

$$I_d = C_{ox}[V_{gs} - V_{th} - V(y)] \cdot \mu \cdot E \cdot W \quad (11)$$

$$(12)$$

$$I_d \cdot dy = \mu \cdot C_{ox} \cdot W[V_{gs} - V_{th} - V(y)] \cdot dV \quad (13)$$

$$(14)$$

$$I_d \int_0^L dy = \mu \cdot C_{ox} \cdot W \int_0^{V_{ds}} [V_{gs} - V_{th} - V(y)] \cdot dV \quad (15)$$

$$(16)$$

$$I_d = \mu \cdot C_{ox} \cdot \frac{W}{L} \left[ (V_{gs} - V_{th}) - \frac{V_{ds}}{2} \right] \cdot V_{ds}$$

- Drain current is the product of charge density and velocity
- Due to the variation in potential from source to drain, charge density must be integrated to determine the drain current
- Let's plot the resulting function...

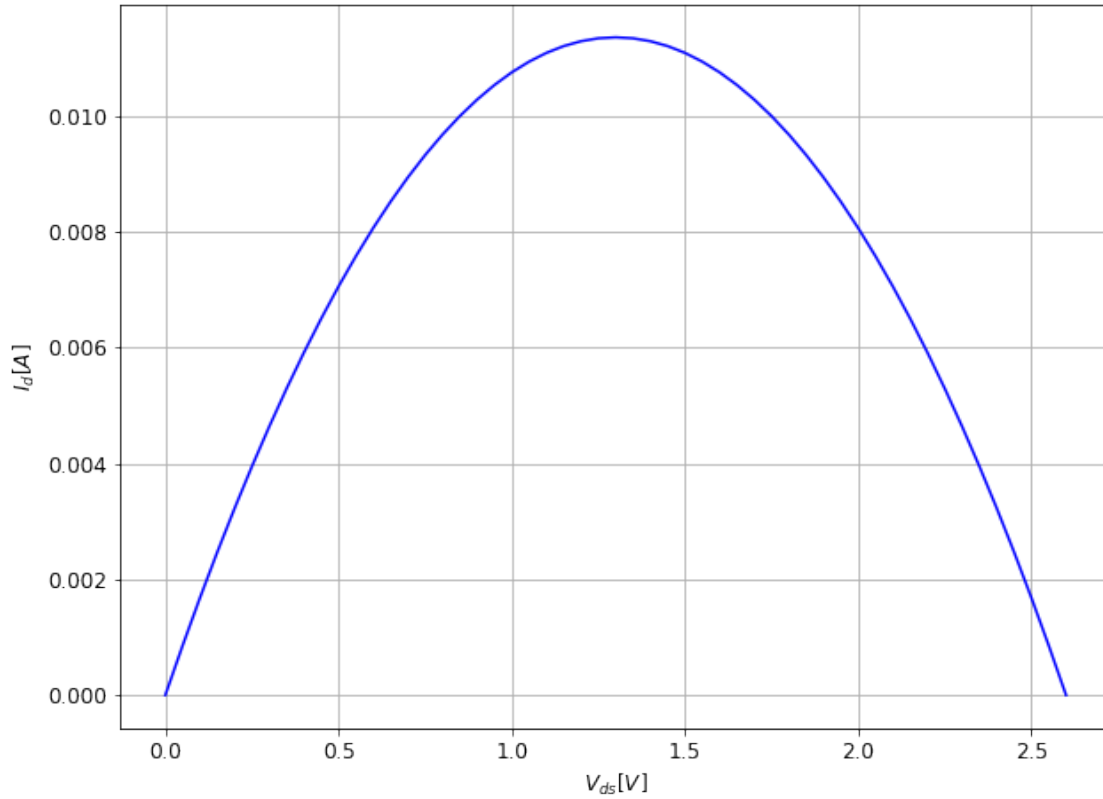
## 2.8 First-order I-V characteristic

```
[3]: u_n = 350                # electron mobility (device parameter)
     e_ox = 3.9*8.854e-12/100; # relative permittivity
     t_ox = 9e-9*100;         # oxide thickness
     C_ox = e_ox/t_ox         # oxide capacitance
     V_th = 0.7               # threshold voltage (device parameter)
     W = 100                  # device width in microns
     L = 1                    # device length in microns

     V_ds = np.arange(0, 2.65, step=0.05)
     V_gs = 2
     I_d = u_n*C_ox*(W/L)*(V_gs - V_th - V_ds/2)*V_ds
```

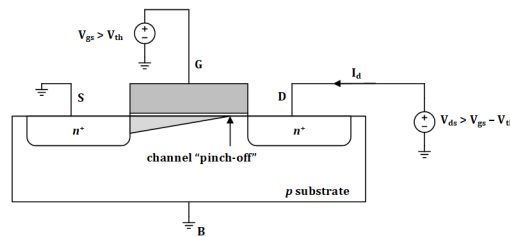
```
[4]: plot_xy(V_ds, I_d, r'$V_{ds}$ [V]', '$I_d$ [A]')
```





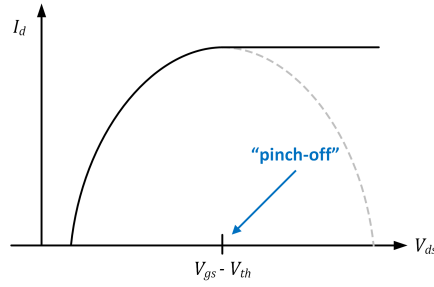
- $I_d$  appears to be decreasing for  $V_{ds} > V_{gs} - V_{th}$
- What really happens when  $V_{ds} > V_{gs} - V_{th}$ ?
- When the potential near the drain region is high enough such that  $V_{gd} = V_{gs} - V_{ds} < V_{th}$ , the inversion charge becomes zero (a phenomenon known as “pinch-off”)

## 2.9 First-order I-V characteristics, revisited



- Near the drain region, charge density is dependent on  $V_{gd}$ , not  $V_{gs}$
- The absence of inversion charge in this region results in high  $E$ -field region, across which the excess  $V_{ds}$  drops
- For  $V_{ds} > V_{gs} - V_{th}$ , drain current becomes “saturated,” no longer increasing with  $V_{ds}$

## 2.10 MOS saturation operation



- In saturation,  $I_d$  is independent of  $V_{ds}$  (to first order), and is given by

$$I_d = \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} - V_{th})^2 \quad (17)$$

- This behavior is what makes the MOS transistor effective as both a current source and a transconductance (gain) element
- What about operation between  $V_{ds} = 0$  and  $V_{ds} = V_{gs} - V_{th}$ ?

## 2.11 MOS triode operation

$$I_d = \mu \cdot C_{ox} \cdot \frac{W}{L} \left[ (V_{gs} - V_{th}) - \frac{V_{ds}}{2} \right] \cdot V_{ds} \quad (18)$$

$$(19)$$

$$= \mu \cdot C_{ox} \cdot \frac{W}{L} \left[ (V_{gs} - V_{th}) \cdot V_{ds} - \frac{V_{ds}^2}{2} \right] \quad (20)$$

- For  $V_{ds} \ll V_{gs} - V_{th}$ :

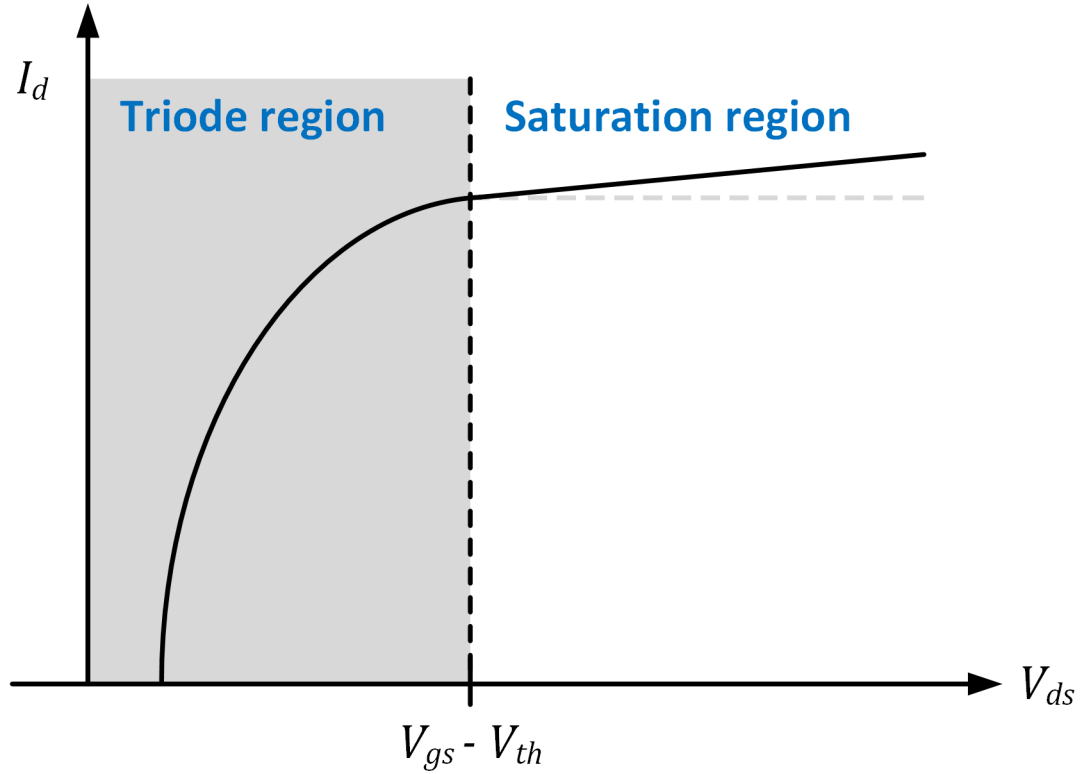
$$I_d \approx \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} - V_{th}) \cdot V_{ds} \quad (21)$$

- For small values of  $V_{ds}$ , drain current is approximately a linear function of  $V_{ds}$
- The MOS transistor in triode can thus be approximated as a resistance:

$$R_{on} = \frac{V_{ds}}{I_d} \approx \frac{1}{\mu C_{ox} \frac{W}{L} (V_{gs} - V_{th})} \quad (22)$$

- Though rarely used as an explicitly resistance, the MOS transistor is modeled this way when operated as a switch

## 2.12 MOS regions of operation



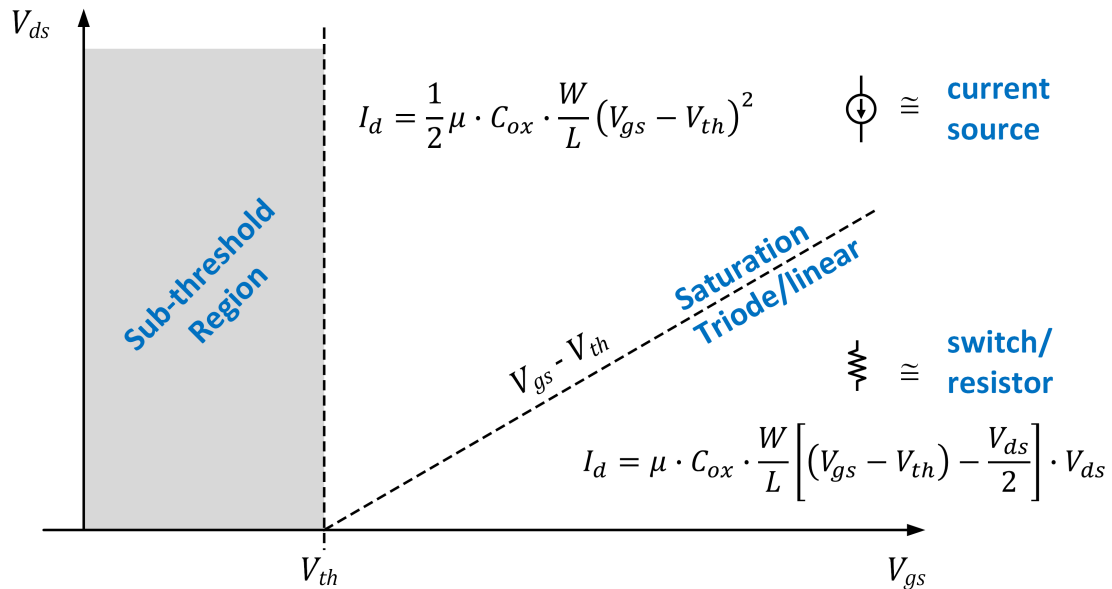
- Triode region expression:

$$I_d \approx \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} - V_{th}) \cdot V_{ds} \quad (23)$$

- Saturation region:

$$I_d \approx \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} - V_{th})^2 \quad (24)$$

## 2.13 First-order MOS model summary



## 2.14 MOS I-V characteristic

```
[69]: def nmos_iv_sweep(V_gs, V_ds, W, L, lmda):
    u_n = 350 # electron mobility (device parameter)
    e_ox = 3.9*8.854e-12/100; # relative permittivity
    t_ox = 9e-9*100; # oxide thickness
    C_ox = e_ox/t_ox # oxide capacitance
    V_th = 0.7 # threshold voltage (device parameter)
    V_ov = V_gs - V_th

    I_d = []

    for i in range(len(V_ds)):
        I_d.append(np.piecewise(V_ds[i], [V_ds[i] < V_ov, V_ds[i] >= V_ov],
                                [u_n*C_ox*(W/L)*(V_gs - V_th - V_ds[i]/
→2)*V_ds[i]*(1+lmda*V_ds[i]) ,
                                0.5*u_n*C_ox*(W/L)*(V_gs - V_th)**2*(1+lmda*V_ds[i])]))

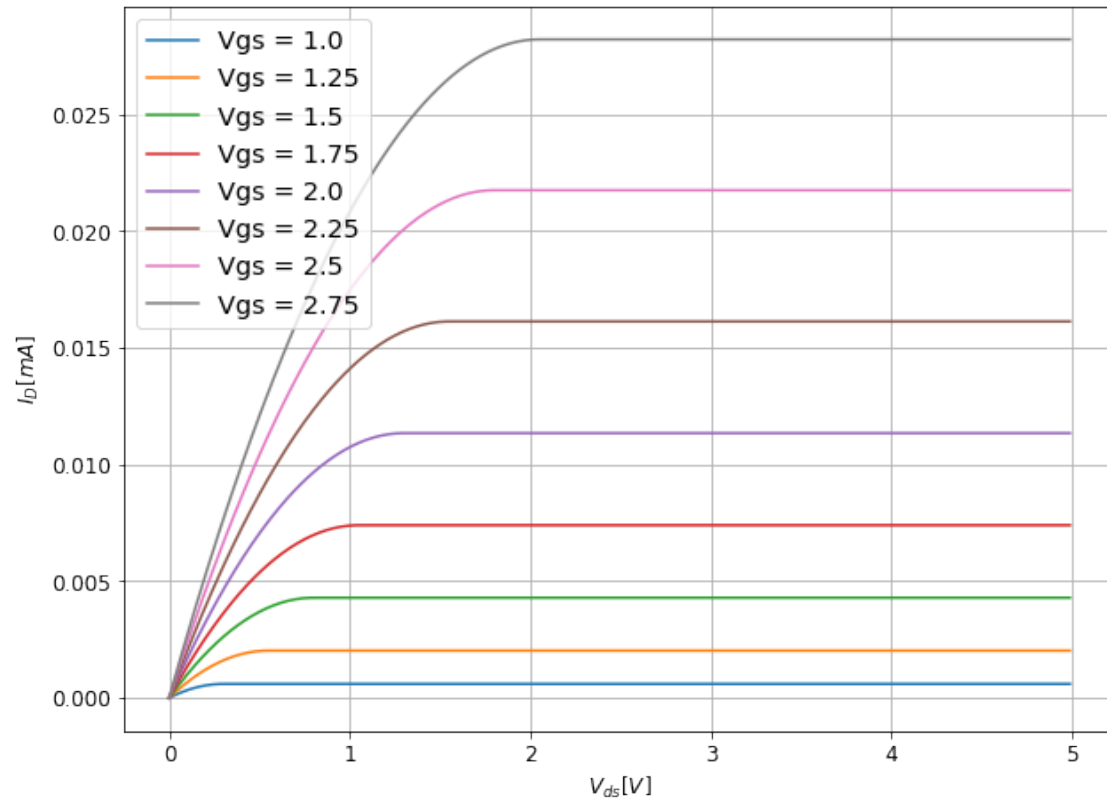
    return np.array(I_d)

def plot_ID_curves(V_gs, V_ds, W, L, lmda):
    fig, ax = plt.subplots(figsize=(10.0,7.5))
    for vgs in V_gs:
        I_d = nmos_iv_sweep(vgs, V_ds, W, L, lmda=lmda)
        ax.plot(V_ds, I_d, label='Vgs = '+ str(vgs))
        ax.set_xlabel(r'$V_{ds}$ [V]$')
        ax.set_ylabel(r'$I_{D}$ [mA]$')
```

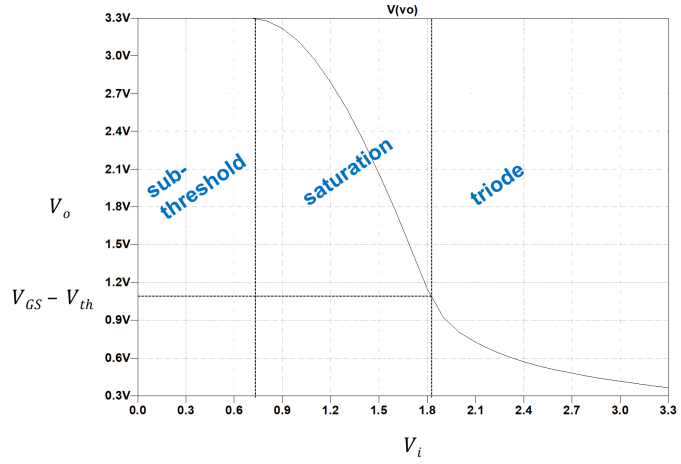
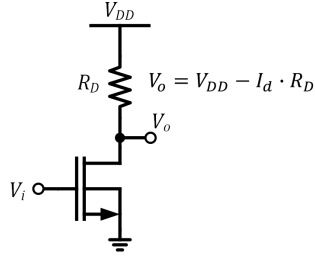
```
ax.grid()  
ax.legend()
```

```
[70]: V_gs = np.arange(1, 3, step=.25)  
V_ds = np.arange(0, 5, step=0.01)  
W = 100  
L = 1
```

```
[71]: plot_ID_curves(V_gs, V_ds, 100, 1, 0)
```



## 2.15 Common-source amplifier (SPICE simulation)



$$I_d = \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_i - V_{th})^2$$

$$V_o = V_{DD} - \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_i - V_{th})^2 \cdot R_D$$

## 2.16 Calculate the gain

$$V_o + \Delta V_o = V_{DD} - \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} + \Delta V_i - V_{th})^2 \cdot R_D \quad (25)$$

$$(26)$$

$$\Delta V_o = -\frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot R_D [(V_{ov} + \Delta V_i)^2 - V_{ov}^2] \quad (27)$$

$$(28)$$

$$= -\frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot R_D [2V_{ov}\Delta V_i + \Delta V_i^2] \quad (29)$$

$$(30)$$

$$= -\frac{2I_d}{V_{ov}} \cdot R_D \Delta V_i \left[ 1 + \frac{\Delta V_i}{2V_{ov}} \right] \quad (31)$$

- The output voltage is quadratic in  $\Delta V_i$ , but a linear approximation is sufficient for most purposes
- Let's use a linear approximation to find the gain...

## 2.17 Small-signal gain

- The input/output relationship is given by

$$\Delta V_o = -\frac{2I_d}{V_{ov}} \cdot R_D \Delta V_i \left[ 1 + \frac{\Delta V_i}{2V_{ov}} \right] \quad (32)$$

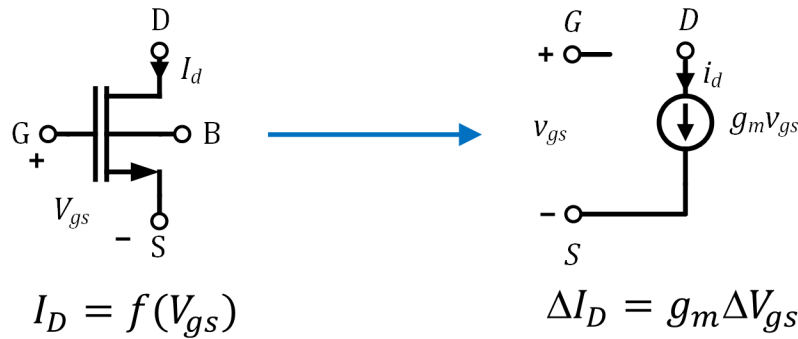
- Assuming  $\Delta V_i \ll 2V_{ov}$ , this becomes

$$\Delta V_o \approx -\frac{2I_d}{V_{ov}} \cdot R_D \Delta V_i \quad (33)$$

- Taking  $\Delta V_i$  to be arbitrarily small, we obtain the “small-signal” gain:

$$A_v = \frac{dV_o}{dV_i} \approx -\frac{2I_D}{V_{ov}} \cdot R_D \quad (34)$$

## 2.18 Small-signal transconductance



- Instead of having to carry out the linearization process for every new circuit we build, we can linearize at the transistor level
- As such, the nonlinear function relating  $I_d$  to  $V_{gs}$  is replaced with a linear transconductance term,  $g_m$
- Transistor linearization allows us to use linear circuit analysis techniques (i.e. Kirchoff's laws) to analyze circuits comprising MOS transistors

### 2.18.1 Linear transconductance versus nonlinear square law

```
[66]: def nmos_iv(V_gs, W, L):
    u_n = 350                                # electron mobility (device parameter)
    e_ox = 3.9*8.854e-12/100;                # relative permittivity
    t_ox = 9e-9*100;                          # oxide thickness
    C_ox = e_ox/t_ox                          # oxide capacitance
    V_th = 0.7                                # threshold voltage (device parameter)

    I_d = 0.5*u_n*C_ox*(W/L)*(V_gs - V_th)**2

    return I_d

def g_m(V_GS0, W, L):
    V_th = 0.7                                # threshold voltage (device parameter)
    I_D0 = nmos_iv(V_GS0, W, L)

    return 2*I_D0/(V_GS0 - V_th)

def gm_line (V_gs, W, L, V_GS0, I_D0):
    return g_m(V_GS0, W, L)*(V_gs - V_GS0) + I_D0
```

```

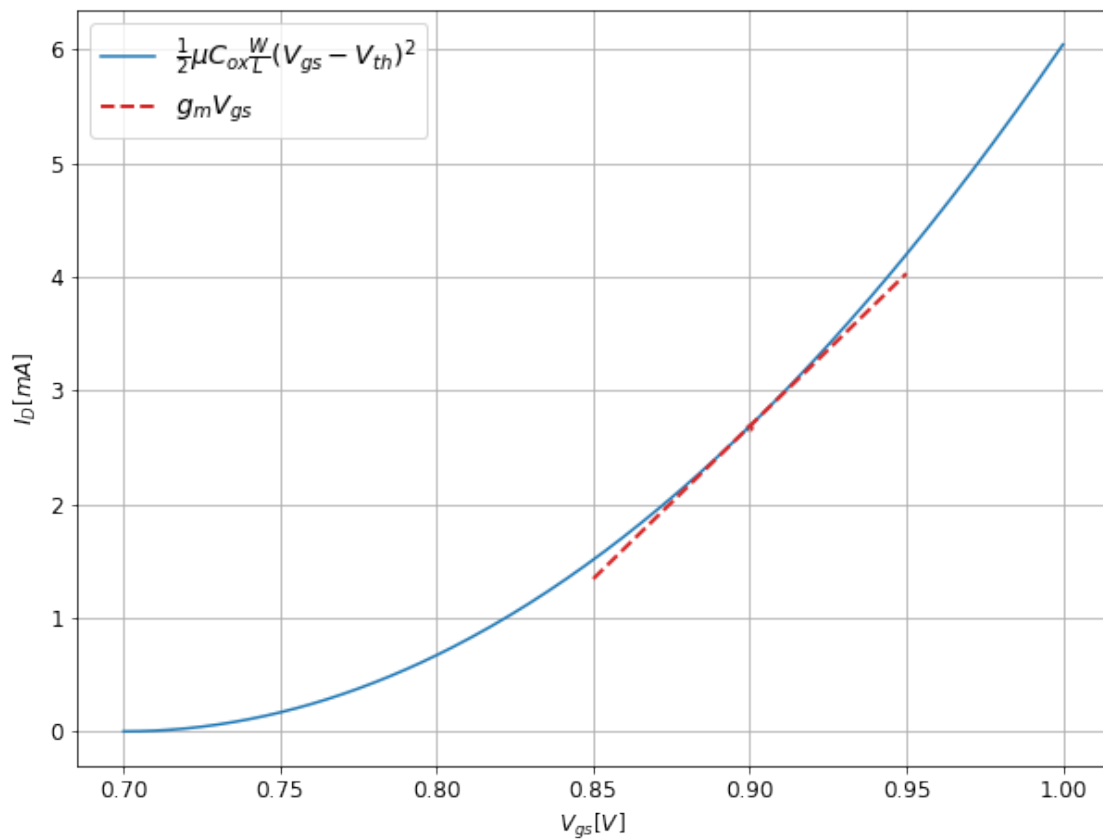
def plot_gm(V_gs, W, L, V_GS1, V_gs_range):
    fig, ax = plt.subplots(figsize=(10.0,7.5))
    ax.plot(V_gs, 1e3*nmos_iv(V_gs, W, L),
            label=r'\frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{gs}-V_{th})^2')
    I_D1 = nmos_iv(V_GS1, W, L)
    ax.scatter(V_GS1, 1e3*I_D1, color='C3',s=10)
    ax.plot(V_gs_range, 1e3*gm_line(V_gs_range, W, L, V_GS1, I_D1),
            'C3--', linewidth=2, label=r'$g_mV_{gs}$')
    ax.set_xlabel(r'$V_{gs}$ [V]')
    ax.set_ylabel(r'$I_{D}$ [mA]')
    ax.grid()
    ax.legend()

```

```

[74]: V_gs = np.linspace(0.7,1,num=300)
V_GS0 = 0.9
V_gs_range = np.linspace(V_GS0-0.05, V_GS0+0.05, 10)
plot_gm(V_gs, 1000, 1, V_GS0, V_gs_range)

```



- In the vicinity of the linearization, or “DC operating” point, the linear model well-approximates the square law



- As  $V_{gs}$  becomes much less or greater than  $V_{GS0}$ , a significant error results in using the linear model
- For this reason,  $g_m$  is used exclusively as a “small-signal” approximation for the transistor behavior

## 2.19 Saturation transconductance

- Transconductance is defined as the derivative of drain current with respect to gate-source voltage

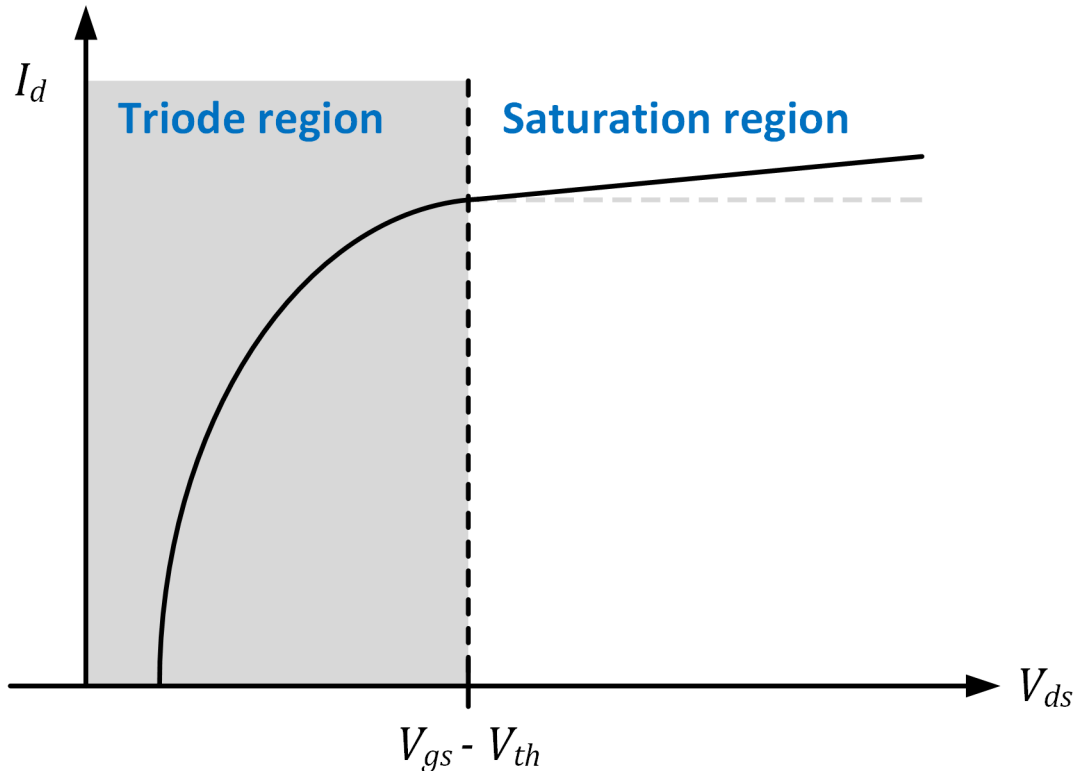
$$g_m = \left. \frac{\partial I_d}{\partial V_{gs}} \right|_{I_d=I_{D0}} = \left. \frac{\partial}{\partial V_{gs}} \frac{1}{2} \mu C_{ox} \frac{W}{L} (V_{gs} - V_{th})^2 \right|_{V_{gs}=V_{GS0}} = \mu C_{ox} \left( \frac{W}{L} \right) (V_{GS0} - V_{th}) \quad (35)$$

- In saturation, this can be expressed as

$$g_m = \frac{2I_{D0}}{(V_{GS0} - V_{th})} = \frac{2I_{D0}}{V_{OV}} \quad (36)$$

- Importantly, for a constant overdrive  $g_m$  is linearly dependent on  $I_{D0}$

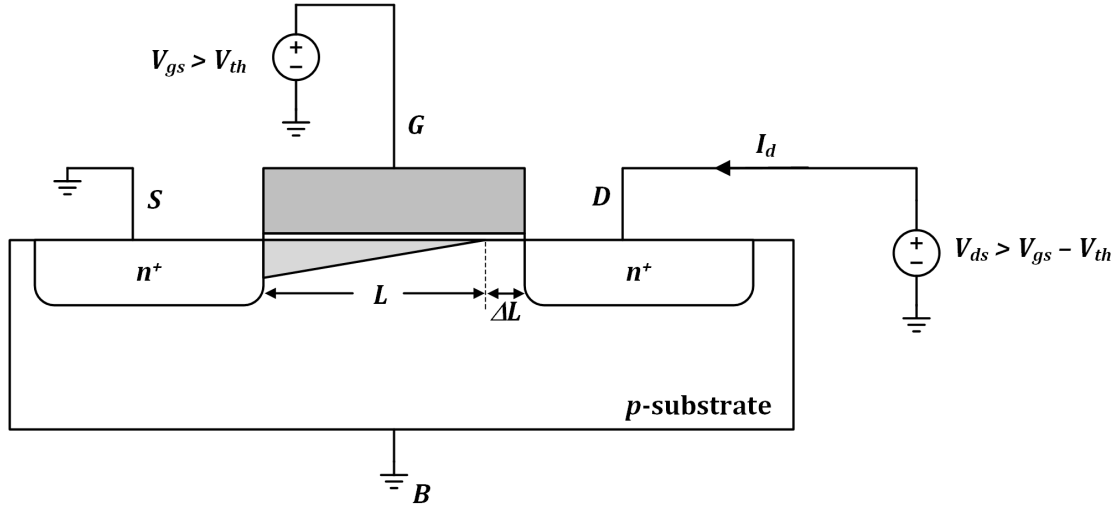
## 2.20 Dependence of $I_d$ on $V_{ds}$



- If we sweep  $V_{ds}$  of a real transistor, we see a small dependence of  $I_d$  on  $V_{ds}$
- The effect is analogous to the Early effect in BJTs

- The variation of  $I_d$  with  $V_{ds}$  is largely due to a decrease in the effective channel length, resulting from the high electric field at the drain
- This effect is typically referred to as “channel-length modulation”

## 2.21 Channel-length modulation



- As  $V_{ds}$  increases, the high  $E$ -field region close to the drain grows, decreasing the effective channel length
- The reduced channel length increases the drain current, resulting in a positive slope of  $I_d$  vs  $V_{ds}$
- The effect is more pronounced for shorter gate lengths, due to the larger ratio  $\Delta L/L$

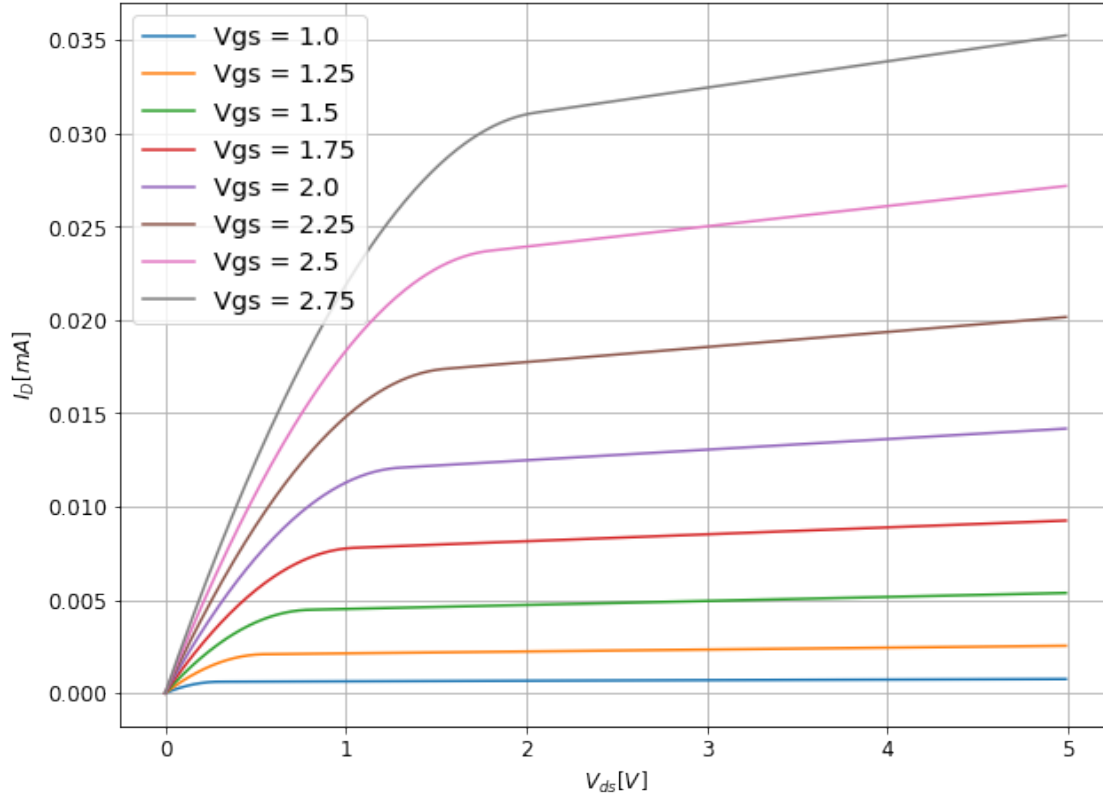
## 2.22 Modified square-law expression

- Finite slope of drain current with respect to  $V_{ds}$  is typically attributed to channel-length modulation
- However, this is actually the result of a combination of a number of physical effects (DIBL, SCBE, ...)
- A simple model that assumes a linear increase in  $I_d$  with  $V_{ds}$  is generally used to model MOS behavior, where the combination of effects is lumped into a single parameter,  $\lambda$ , inversely proportional to channel length  $L$  ( $\lambda \propto 1/L$ )
- The drain current expression is modified as

$$I_d = \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} - V_{th})^2 (1 + \lambda V_{ds}) \quad (37)$$

## 2.23 I-V characteristic with channel-length modulation

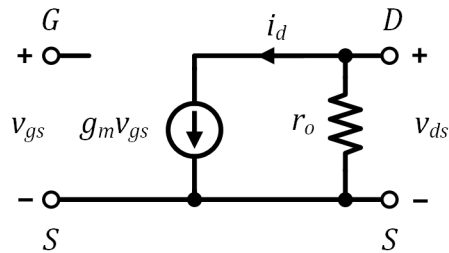
[73]: `plot_ID_curves(V_gs, V_ds, 100, 1, .05)`



- The inclusion of  $\lambda$  in the square-law model results in a finite slope of the  $I_d$  vs  $V_{ds}$  curves
- This dependence of  $I_d$  on  $V_{ds}$  is referred to as the transistor *output resistance*, which is determined by taking the derivative of  $I_d$  with respect to  $V_{ds}$ :

$$r_o = \left( \frac{d}{dV_{ds}} I_d \right)^{-1} \Big|_{I_d = I_{D0}} = \frac{1}{\lambda \cdot \frac{1}{2} \mu \cdot C_{ox} \cdot \frac{W}{L} (V_{gs} - V_{th})^2} \Big|_{V_{gs} = V_{GS0}} = \frac{1}{\lambda I_{D0}} \quad (38)$$

## 2.24 MOS DC model (intrinsic gain)



$$g_m = \frac{2I_{D0}}{V_{OV}} \quad (39)$$

$$r_o = \frac{1}{g_{ds}} = \frac{1}{\lambda I_{D0}} \quad (40)$$

$$\left| \frac{v_{ds}}{v_{gs}} \right| = \frac{g_m}{g_{ds}} = \frac{2}{V_{OV}} \quad (41)$$

- The dependence of  $I_d$  on  $V_{ds}$  is modeled as a resistance  $r_o$ , referred to as the transistor *output impedance*, which is linearly dependent on  $I_d$
- The product  $g_m \cdot r_o$  is called the intrinsic gain of the MOSFET, which is, to first order, independent of drain current
- As we will see, intrinsic gain is a fundamental device property that affects a number of circuit performance metrics (e.g. voltage gain of an amplifier)

## 2.25 Level 1 NMOS/PMOS SPICE models

NMOS Model			
LEVEL = 1	VTO = 0.7	GAMMA = 0.45	PHI = 0.9
NSUB = 9e+14	LD = 0.08e-6	UO = 350	LAMBDA = 0.1
TOX = 9e-9	PB = 0.9	CJ = 0.56e-3	CJSW = 0.35e-11
MJ = 0.45	MJSW = 0.2	CGDO = 0.4e-9	JS = 1.0e-8

PMOS Model			
LEVEL = 1	VTO = -0.8	GAMMA = 0.4	PHI = 0.8
NSUB = 5e+14	LD = 0.09e-6	UO = 100	LAMBDA = 0.2
TOX = 9e-9	PB = 0.9	CJ = 0.94e-3	CJSW = 0.32e-11
MJ = 0.5	MJSW = 0.3	CGDO = 0.3e-9	JS = 0.5e-8

## 2.26 Comments on models

- The model discussed here is often referred to as the “long-channel” or “square-law” model
- Although it is intuitively satisfying and readily applied analytically, it is grossly inadequate for modeling modern “short-channel” devices
- To accurately model a wide range of “second-order” effects, a complex transistor model with many empirical parameters, must be used
- However, the long-channel model is useful for gaining intuition and understanding general performance trends
- A rule of thumb is useful here: **Use the simplest model that is accurate enough for the task**

## 2.27 Summary

- Long-channel “square-law” model provides an intuitive picture of MOS device operation
  - $g_m$  linearly dependent on  $I_d$  if  $V_{gs} - V_{th}$  is constant
  - $r_o$  inversely dependent on  $I_d$
  - Intrinsic gain  $g_m \cdot r_o$  independent of  $I_d$

- Small-signal model will be used to design circuits with specific gain and impedance characteristics