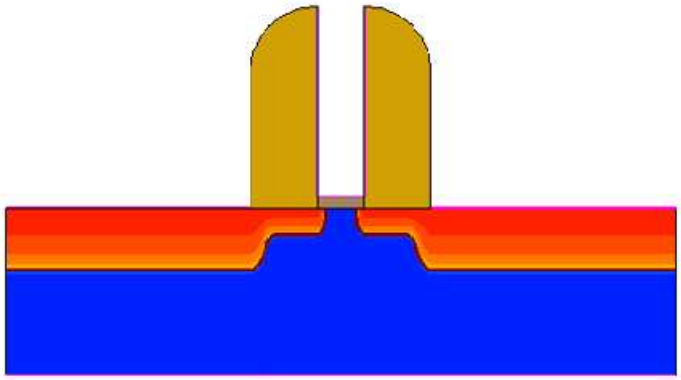
Project 2

MOSFET

EECS 3610 Semiconductor Physics and Devices

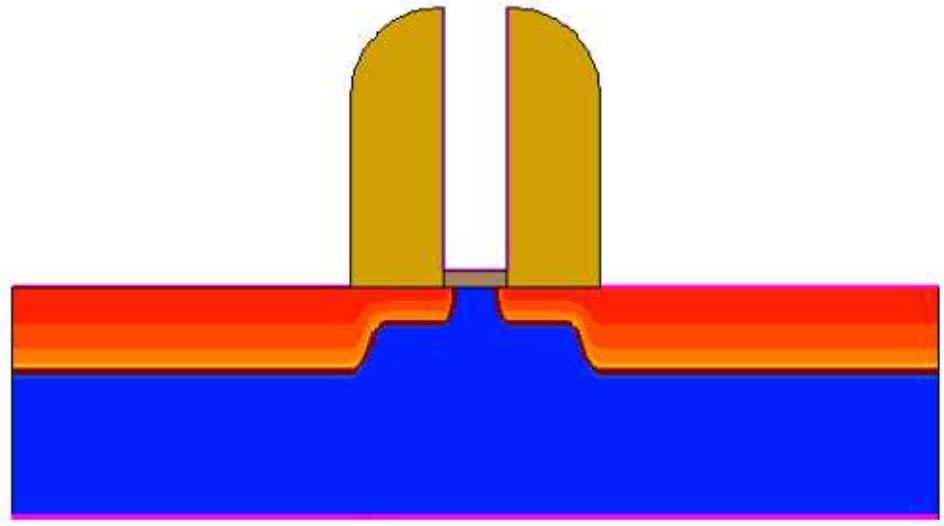
Fall 2025

Professor Gerd Grau



1 Project 2: MOSFET

In this project, you are going to simulate a short-channel NMOSFET. You will simulate it according to specifications given by the International Technology Roadmap for Semiconductors (ITRS) at the 2017 node with a gate length of 13.9 nm. This was the last node where planar bulk MOSFETs were specified. You need to achieve both the specified on- and off-current i.e. the maximum and minimum currents at V_{gs}=0 and V_{gs}=V_{dd} when V_{ds}=V_{dd}. The challenge in a short-channel MOSFET is that the gate only has limited control over the channel, especially the back of the channel. If this is not countered, the off-current will be large. Your short-channel MOSFET will need to contain a few features to do so as displayed in the below cross-section.



ITRS Specifications:

Variable	Specification
MPU/ASIC Metal 1 (M1) ½ pitch (nm)	25.3
L_g: Physical Lgate for HP logic (nm)	13.9
V_{dd}: Power Supply Voltage (V)	0.80
EOT: Equivalent Oxide Thickness (nm)	0.67
Oxide dielectric constant	14.5
Physical oxide thickness (nm)	2.49
Channel Doping (10^{18} cm^{-3})	9.0
$I_{d,leak}$ (nA/µm)	100
$I_{d,sat}$: NMOS Drive Current (µA/µm)	600

The source and drain regions each consist of two n-doped regions of different depth. Close to the channel, the source and drain are shallow to prevent conduction far away from the gate. These features are called extensions. Away from the channel, the source and drain are deeper to minimize series resistance. Both regions are fabricated by ion implantation. First, the extensions are created by a shallow implantation. Then, dielectric spacers (yellow regions e.g. Si_3N_4) are created on top of the extensions to block the dopants during the second ion implantation step. In your simulation, you need to optimize the depth and the length of the extensions. The following parameters are fixed:

- A Gaussian doping profile describes the ion-implanted dopants. Use the analytical doping profile functionality in Structure Editor.
- The peak doping density for both implantations is $2*10^{20}$ cm^{-3}
- The depth of the deep regions is 20 nm. Junction depth is defined as where the donor and acceptor doping densities are equal.
- The metal half-pitch (25.3 nm) is the distance from the center of the source (or drain) contact to the center of the gate. The physical gate length (13.9 nm) is the width of the gate contact. You can vary the length of the extensions as long as the half-pitch and gate length have these values.

Another important technique is retrograde doping. A larger p-type doping density is used further away from the channel. The doping density inside the channel determines the threshold voltage at which the transistor turns on and affects mobility in the channel. The channel doping density of $9.0*10^{18}$ cm^{-3} is set by the ITRS. By increasing doping below the channel, you can reduce leakage along this current path. Add a Gaussian doping profile of Boron below the source and drain electrodes and the channel.

In order to achieve good gate control, we need a very thin gate oxide. SiO_2 would be so thin that it would not block the gate current anymore. Therefore, we are going to use a high-k gate dielectric i.e. a different oxide material with a higher dielectric constant. When you create your structure, choose HfO_2 for the gate dielectric. With a dielectric constant of 14.5, you can use a dielectric thickness of 2.49 nm, which has the same capacitance as a SiO_2 layer of thickness 0.67 nm, called equivalent oxide thickness (EOT). In order to tell SDevice the dielectric constant ε_r of HfO_2, follow these steps:

1) Right click on the SDevice icon > Edit Input > Parameter
2) Choose material HfO_2
3) Enter the following in the parameter file:

```
#define ParFileDir .

Material="HfO2" {
  #includeext "ParFileDir/HfO2.par"
```

```
        Epsilon
        {
                epsilon= 14.5  # [1]
        }
}
```

4) In the sdevice_des.cmd file, make sure your file clause contains the following line:

```
Parameter="@parameter@"
```

As a gate material, we are going to use a metal, which prevents any problems associated with poly-Si gates such as depletion of the gate. We will simply model the metal gate as an electrode and don't need to include it in our simulation as a separate material. We can model a particular metal by specifying its work function when defining the electrode in SDevice. Many different metals can be used as a metal gate. You can use this parameter to tune the threshold voltage and shift your I_d-V_g-curve left or right. A good starting point is a work function of 4.3 eV.

```
{ Name="gate"      Voltage= 0.0 WorkFunction = 4.3}
```

Notice the body electrode at the bottom of the transistor model. Include it in Structure Editor and bias it with 0V in SDevice.

Some other modifications you need to make to your project compared to the diode project:

Use the following Physics models in SDevice that capture the relevant physics in a MOSFET:

```
Physics{
  EffectiveIntrinsicDensity( OldSlotboom )
}

Physics(Material="Silicon"){
  MLDA
  Mobility(
    PhuMob
    HighFieldSaturation
    Enormal
    DopingDependence
  )
  Recombination(
    SRH( DopingDependence )
    Band2Band
```

```
  )
}
```

You probably need to increase the number of iterations in the Math section if your simulation does not complete. Another factor can be the mesh size and the step size of the voltage sweep.

In a transistor, there are two input voltages, V_g and V_d. Typically, one is swept while keeping the other constant. For example, for an I_d-V_g-curve, SDevice first ramps up V_d and then keeps it constant while sweeping V_g. Consider the below code. You can use this code for I_d-V_g-curves and adapt it to create I_d-V_d-curves. It is common to do two I_d-V_g-curves with different values of V_d (linear and saturation) and five I_d-V_d-curves with different values of V_g. For each curve, create a different node on the workbench and combine them in a single Inspect plot.

```
Solve {
*- Creating initial guess:
  Coupled(Iterations= 100 LineSearchDamping= 1e-4){ Poisson }
  Coupled { Poisson Electron }

 *- Ramp drain to Vd
  Quasistationary(
    InitialStep= 1e-1 Increment= 1.35
    MinStep= 1e-5 MaxStep= 0.5
    Goal { Name="drain" Voltage=@Vds@ }
  ){ Coupled { Poisson Electron } }

*- Vg sweep
  NewCurrentFile="IdVg_"
  Quasistationary(
    DoZero
    InitialStep= 1e-3 Increment= 1.35
    MinStep= 1e-5 MaxStep= 0.04
    Goal { Name="gate" Voltage= @Vgs@ }
  ){ Coupled { Poisson Electron }
    CurrentPlot( Time=(Range=(0 1) Intervals= 30)  )
  }
}
```

Set up Inspect to automatically extract the important device metrics on-current, off-current, threshold voltage (V_t), subthreshold swing (SS). Utilize the built-in functions.

Deliverables:

Write a **5-page** report double-column journal style using the IEEE template on eClass. Your report needs to include the following:

- A table with the input parameters of your optimized transistor and output device performance, namely:
  - Extension junction depth
  - Extension length
  - Retrograde peak doping density
  - Retrograde doping peak position
  - Gate work function
  - On-current
  - Off-current
  - $V_{t,lin}$
  - $V_{t,sat}$
  - Subthreshold swing
- A plot of your optimized device structure including mesh
- An I_d-V_g-graph of your optimized structure both with the current on a logarithmic scale and of $I_d^{1/2}$ on a linear scale
- An I_d-V_d-graph of your optimized structure on a linear scale
- A discussion of the effect of the different input parameters on device performance. Include any plots or graphs that support your arguments.

In addition to your report, upload your simulation files (your project folder) to eClass. Create one document that combines all your .cmd files for SDE, SDevice and Inspect.

Due: **December 14th 2025**