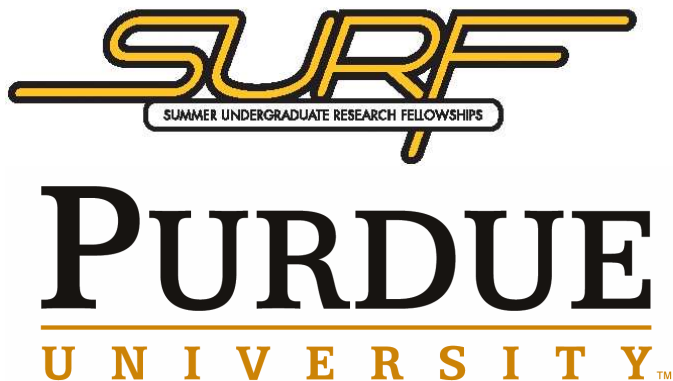




Network for Computational Nanotechnology (NCN)

Purdue, Norfolk State, Northwestern, MIT, Molecular Foundry, UC Berkeley, Univ. of Illinois, UTEP



nanoFET: Upgrading to NEMO5

Kelsey Holley, Seung Hyun Park, Gerhard Klimeck

Network for Computational Nanotechnology (NCN)
Electrical and Computer Engineering



Semiconductor
Research
Corporation

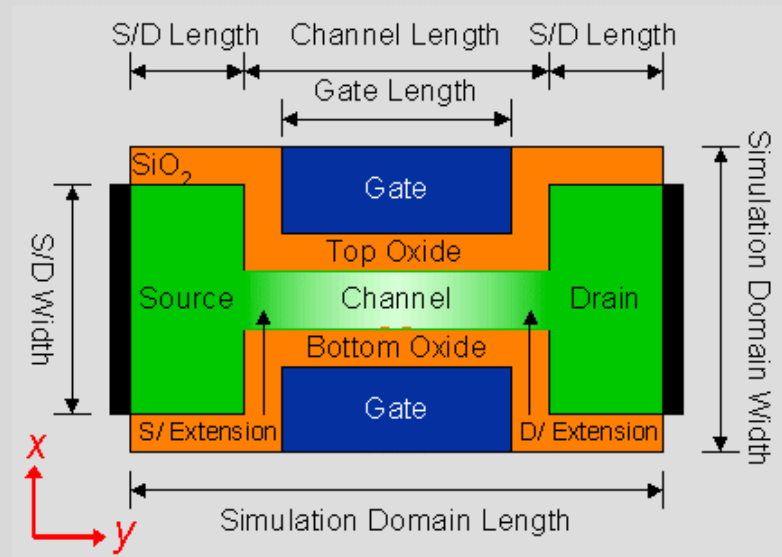


holleyk@purdue.edu

- Introducing nanoFET
- Software & Simulations
- Working with Inputs
- Current Status
- Future Work
- Questions

- Overview
 - Simulates ballistic transport in 2D, double-gated MOSFET devices
 - Four phases of inputs
- Objective
 - Upgrade tool using newly developed software

1 Device Settings → 2 Material Settings → 3 Bias Settings → 4 Simulator St



Dimensions | Mesh/Grid | Doping

Simulation Domain Length: **29nm**

Simulation Domain Width: **24nm**

Source/Drain Length: **6nm**

Source/Drain Width: **3nm**

Channel Length: **17nm**

Material Settings >

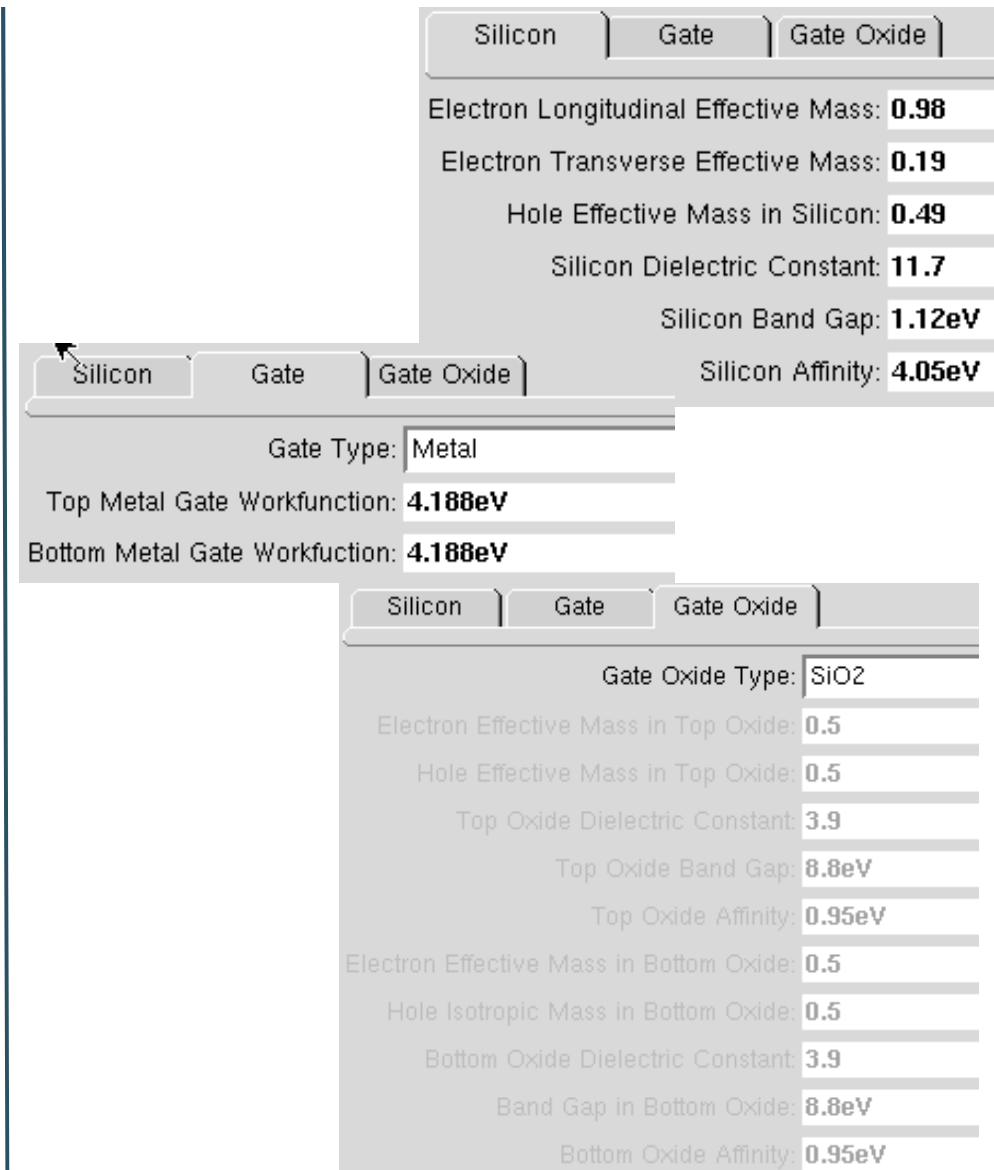
- Dimensions
 - Simulation domain
 - Source/Drain
 - Channel
 - Gate
 - Oxide
- Mesh & Grid
 - No. of grid
 - Grid Size along channel
- Doping
 - Impurity atoms per cm^3

Dimensions	Mesh/Grid	Doping
Simulation Domain Length:	29nm	
Simulation Domain Width:	24nm	
Source/Drain Length:	6nm	
Source/Drain Width:	3nm	
Channel Length:	17nm	
Channel Width/Thickness:	3nm	
Gate Length:	9nm	
Top Oxide Thickness:	1nm	
Bottom Oxide Thickness:	1nm	

Dimensions	Mesh/Grid	Doping
No. of Grid (nx) in Top Oxide:	5	
No. of Grid (nx) in Channel:	15	
No. of Grid (nx) in Bottom Oxide:	5	
Grid Size Along Channel (deltaY):	0.3nm	

Dimensions	Mesh/Grid	Doping
Source/Drain Doping (/cm ³):	2.e20	
Source/Drain Extension Doping Type:	Gaussian	
Gaussian Parameter, g (db/nm):	1	
S/D Extension Abrupt Doping (/cm ³):	2.e20	
Gate Polysilicon Doping (/cm ³):	2.e20	
Channel/Body Doping (/cm ³):	-0.e20	

- Silicon
 - Effective Masses
 - Dielectric Constant
 - Band Gap
 - Affinity
- Gate
 - Metal or n-Polysilicon
- Gate Oxide
 - SiO₂ or Other Oxide
 - Must specify Other Oxide Parameters



The screenshot displays the 'Material Settings' interface with three tabs: Silicon, Gate, and Gate Oxide. The Silicon tab is active, showing the following parameters:

- Electron Longitudinal Effective Mass: **0.98**
- Electron Transverse Effective Mass: **0.19**
- Hole Effective Mass in Silicon: **0.49**
- Silicon Dielectric Constant: **11.7**
- Silicon Band Gap: **1.12eV**
- Silicon Affinity: **4.05eV**

The Gate tab is also visible, showing:


- Gate Type: **Metal**
- Top Metal Gate Workfunction: **4.188eV**
- Bottom Metal Gate Workfunction: **4.188eV**

The Gate Oxide tab is visible but dimmed, showing:

- Gate Oxide Type: **SiO₂**
- Electron Effective Mass in Top Oxide: **0.5**
- Hole Effective Mass in Top Oxide: **0.5**
- Top Oxide Dielectric Constant: **3.9**
- Top Oxide Band Gap: **8.8eV**
- Top Oxide Affinity: **0.95eV**
- Electron Effective Mass in Bottom Oxide: **0.5**
- Hole Isotropic Mass in Bottom Oxide: **0.5**
- Bottom Oxide Dielectric Constant: **3.9**
- Band Gap in Bottom Oxide: **8.8eV**
- Bottom Oxide Affinity: **0.95eV**

- Environment
 - Temperature of environment around device
- Bias (applied voltage)
 - IdVg (gate controlled), IdVd (drain controlled), or single bias
 - Incremental gate voltage determines step size between bias points
 - Gate voltage controls both top and bottom gate
- Energy
 - Uniform or nonuniform grid

Environment | Bias | Energy

Ambient Temperature:  300K

Environment | Bias | Energy


Simulation For: IdVg

IdVg Inputs

Source Voltage:	0V
Drain Voltage:	0.4V
Initial Gate Voltage:	-0.2V
Final Gate Voltage:	0.4V
Incremental Gate Voltage:	0.05V

Environment | Bias | Energy

Energy Grid (uniform/nonuniform): nonuniform

Total Egrid Size:  512

Lower Limit of Energy: -0.4eV

Higher Limit of Energy: 0.8eV

Include Resonance in the Active Region?: yes

- Simulator Options
 - Full Simulation or Flat Potential
- Simulator Parameters
 - Sets up simulation time

Simulation Options | Simulator Params

Simulation Mode: FullSimulation

Solve Poisson?: yes

No. of Poisson Iterations:

Include Electron Mass Anisotropy?: yes

Calculate LessThan Green's Function?: yes

Calculate GreaterThan Green's Function?: yes

Do a Backward Recursion?: yes

Depth for 1D-Plot Along Length: 1nm

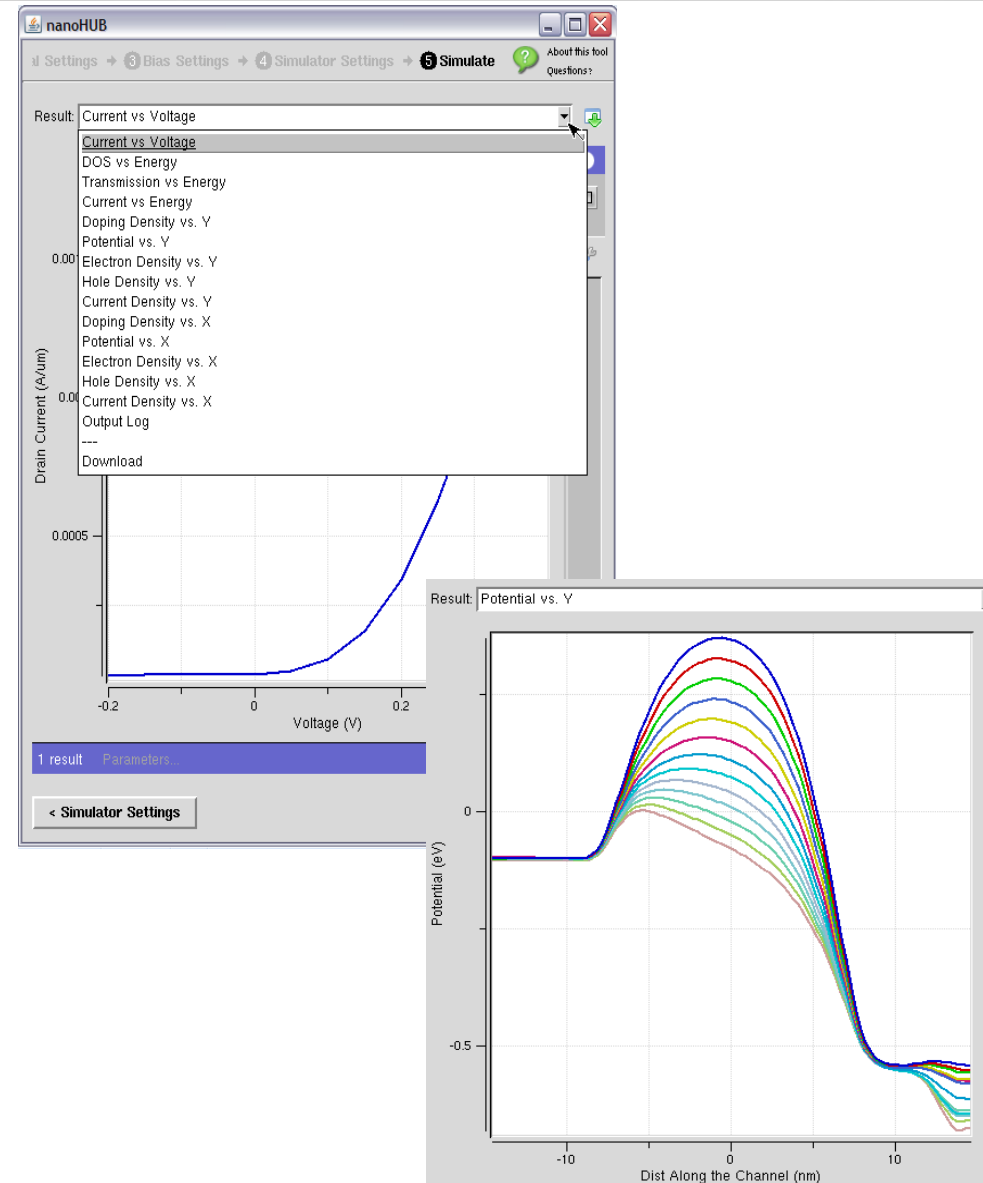
Length for 1-D Plot Along Depth: 14.5nm

Simulation Options | Simulator Params

Estimated Run Time: 02:00:00

Format is hh:mm:ss (hh = hour, mm = minutes, ss)

- 14 plots
 - Current vs. Voltage
 - Density Of States vs. Energy
 - Current vs. Energy
 - Doping Density vs. Y, vs. X
 - Potential vs. Y, vs. X
 - Electron Density vs. Y, vs. X
 - Hole Density vs. Y, vs. X
 - Current Density vs. Y, vs. X
- Output Log
- Plots are downloadable



Rappture

- **Rapid Application Infrastructure**
- Can be used with C, C++, Fortran, MATLAB, Octave, Perl, Python, and Tcl.
- Creates graphical user interface (GUI)
- Two files:
 - XML tool description
 - Launcher file

NEMO5

- **NanoElectronics MOdeling Tools**
- Klimeck research group at Purdue
- Two input files:
 - Input deck
 - Material database file

Parts of a Simulation: Rappture

tool.xml

- Extensible markup language
- Input/output description
- Lists characteristics of each parameter

```
- <number id="ChannelLength">
- <about>
  <label>Channel Length</label>
  <description>Channel Length is considered to be the
  total distance in between the Source and Drain
  regions (Channel Length = 2*(S/D Extension
  Length) + Gate Lengh).</description>
  <color>green</color>
</about>
<units>nm</units>
<min>1nm</min>
<max>100nm</max>
<default>17.0nm</default>
</number>
- <number id="ChannelWidth">
- <about>
  <label>Channel Width/Thickness</label>
  <color>green</color>
```

Launcher file

- User chooses language
- Three parts
 - Inputs from GUI
 - Algorithm
 - Results back to GUI

```
// Get the current paths (equivalent to UNIX pwd)
char input_dir[1000];
char *cwd=getcwd(NULL,0);
sprintf(input_dir,"%s",cwd);

char output_dir[1000];
sprintf(output_dir,".");

printf("Input directory: %s\n", input_dir);
printf("Output directory: %s\n", output_dir);

/******
  READ INPUT FROM lib & STORE IN VARIABLES
******/
```

Input Deck

- Specifies material/device and simulation information
- Contains three sections
 - Structure, Solvers, Global

```
Structure
{
  Material
  {
    tag = channel

    name = Si
    crystal_structure = simplecubic
    regions = (2)
    Lattice:a_lattice = 0.300

    Bands:BandEdge:mstar_c_dos = 0.980
  }

  Material
  {
    tag = lead

    name = Si
  }
}
```

Material Database

- Specifies material parameters
- File provided by NEMO5 developers

```
# Si from Si.mat substrate file

group Si {
  unit_cell = 0.27155;
  type      = "IV";
  group Lattice {
    variables      = "Tkel";
    a_lattice      = 0.54310;
    #a_lattice     = 0.543; #Mathieu
    epsilon_dc     = 11.9;
    c11            = 1.675;
    c12            = 0.650;
    c44            = 0.801;
    c_elastic     = cp_3to1(c11,c12,c44);
    bdp            = -2.1;
    ddp            = -4.8;

    element       = Si;
    Si_mass       = 28.0855;
    Si_mass_ref   = "NIST periodic table";

    #strain_alpha  = 48.5;
    #strain_beta   = 13.8;
    #strain_ref    = "Martin, PRB 1 (1970)";
  }
}
```

- Used same tool.xml
- Launcher file
 - Completed reading inputs from GUI

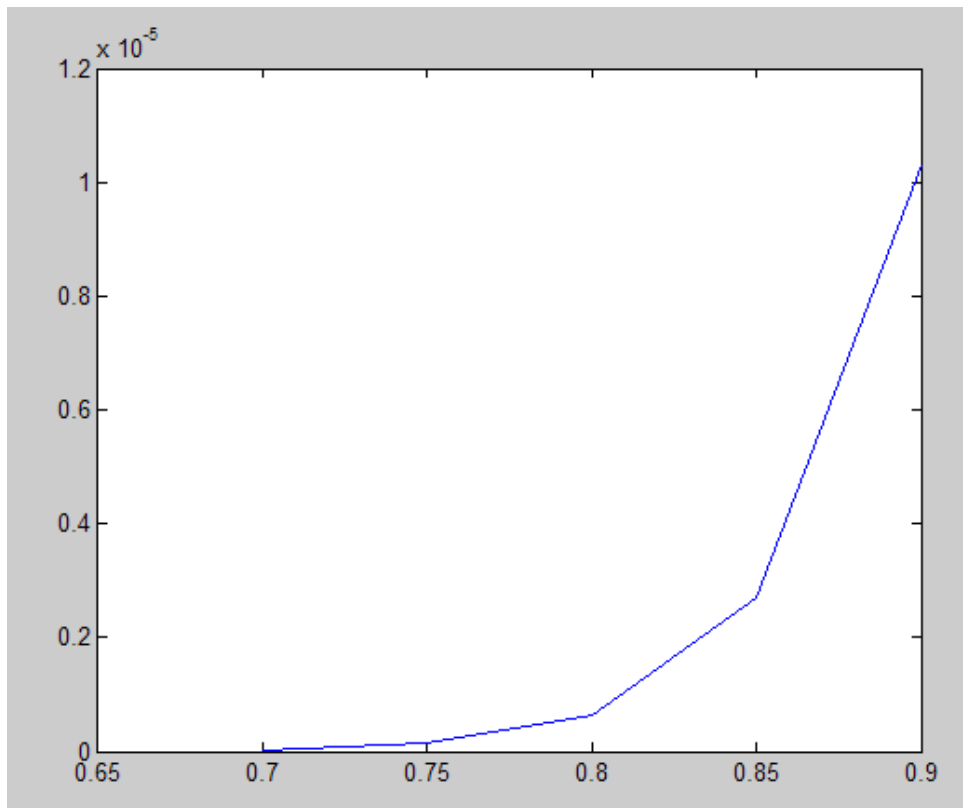
```
//Phase: Device Settings, Group: Dimensions
rpGetString(lib, "input.phase(DevPhase).group(param).number(Ly).current", &data);
Ly = rpConvertDbl(data, "nm", &err);
if (err) {printf("Error while retrieving Simulation Domain Length."); exit(1);}
```

- Working on printing input deck
- Working with developers
- Currently, 50% of inputs completed

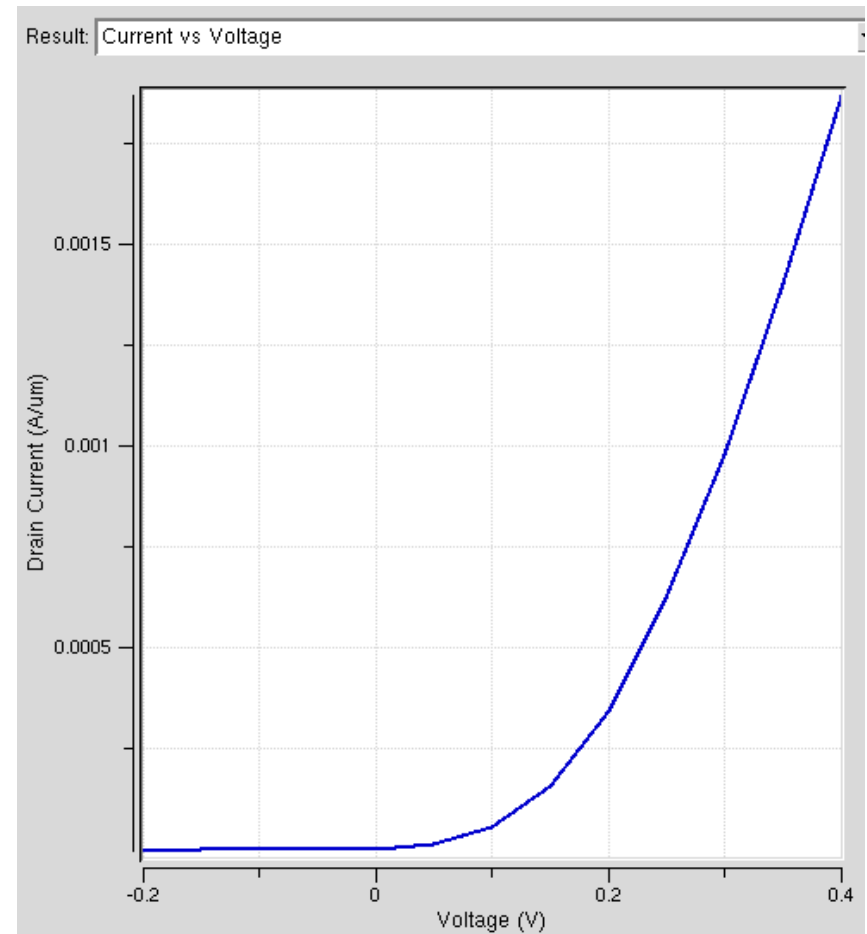
```
//STRUCTURE

//Structure/Material/Channel
fprintf(fpindeck, "Structure\n{\n\tMaterial\n\t{");
fprintf(fpindeck, "\n\t\ttag = channel\n\n\t\tname = Si\n\t\t\tcrystal_structure = simplecubic\n\t\t\tregions = (2)"); //sections that
won't change (tag, name, crystal structure, and regions)
fprintf(fpindeck, "\n\t\tLattice:a_lattice = %.3f\n\t}", GridSizeY);
//lattice
```

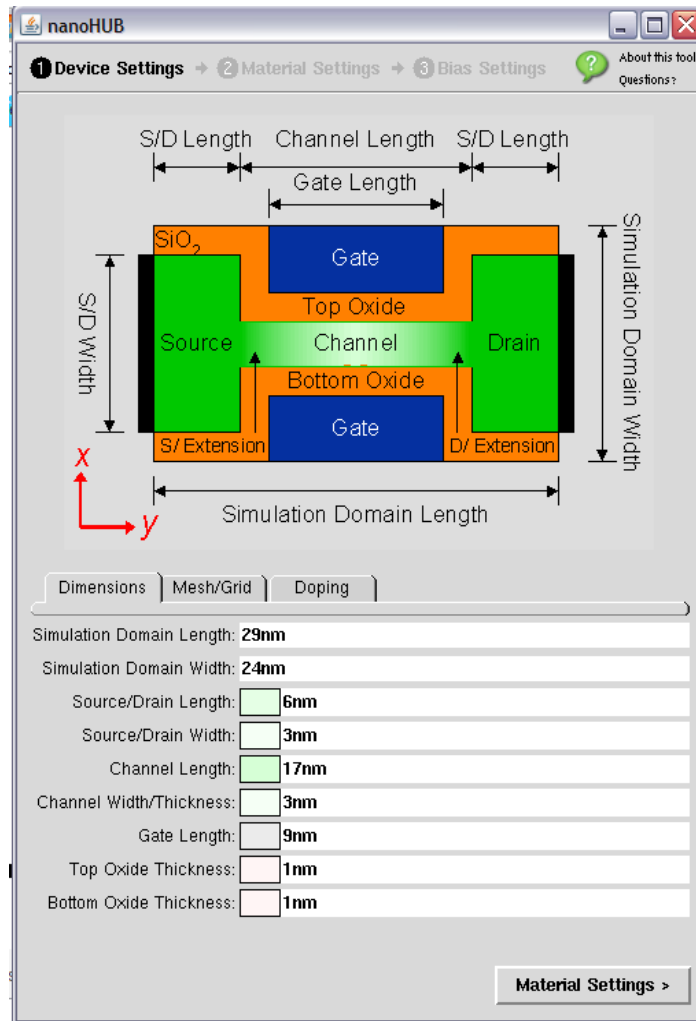
Current Output



Desired Output



- To finish tool
 - Incorporate or remove remaining inputs
 - Run NEMO5 simulation
 - Resulting data must be read back into the launcher file
 - Graphs created and sent to GUI
- Testing of tool
- Create user guide
- Launch on nanoHUB



Questions?