nanoHUB.org
online simulations and more

Computational Electronics

Introduction to Silvaco ATLAS

Prepared by
Dragica Vasileska
Associate Professor
Arizona State University

nanoHUB.org
online simulations and more

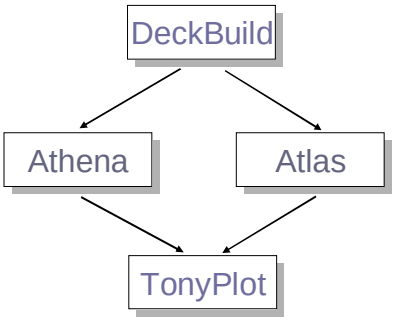
Introduction to Silvaco ATLAS

1. Some general comments
2. Deckbuild overview
3. ATLAS syntax
   - (A) Structure specification
   - (B) Materials models specification
   - (C) Numerical method selection
   - (D) Solution specification
   - (E) Results analysis
4. ATLAS Extract description
5. Examples
   - (A) Diode example

nanoHUB.org
online simulations and more

Some General Comments

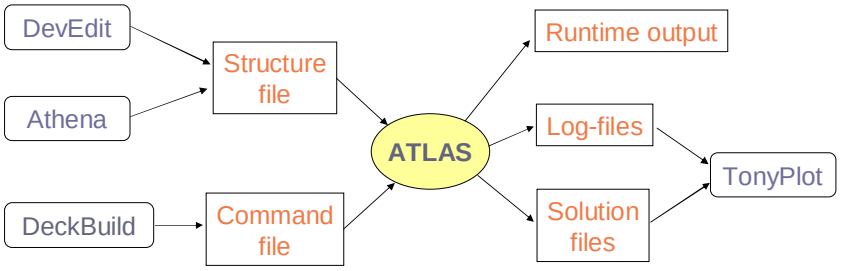
- **The VWF** (Virtual Wafer Fab) Framework consists of two different sets of tools:
  - core tools
  - auxiliary tools
- **ATHENA** - process simulation tool
  - predicts the physical structure that results from the processing steps
  - treats process simulation as a serial flow of events
- **ATLAS** - device simulation tool
  - performs physically-based 2D/3D device simulations
  - predicts the electrical behavior of specified semiconductor structures and provides insight into the internal physical mechanisms associated with the device operation
  - various tools that comprise ATLAS include: S-PISCES, BLAZE, GIGA, TFT, LUMINOUS, LASER, MIXEMODE, DEVICE3D, INTERCONNECT3D, THERMAL3D



nanoHUB.org
online simulations and more

ATLAS Inputs and Outputs

- Most ATLAS simulations use two types of inputs: **text files** and **structure files**
- There are three types of outputs produced by ATLAS:
  1) ***Runtime output*** - guide to the progress of simulation that is running
  2) ***Log files*** - summaries of the electrical output information
  3) ***Solution files*** - store 2D and 3D data relating to the values of the solution variables



nanoHUB.org
online simulations and more

Modes of Operation

There are three different modes of operation of ATLAS:

1) Interactive mode with DeckBuild

deckbuild -as <input_filename>

2) Batch mode with DeckBuild

With X-Windows operation:

deckbuild -run -as <input_filename> -outfile <output_filename>

Without X-Windows operation:

deckbuild -run -ascii -as <input_filename> -outfile <output_filename>

3) Batch mode without DeckBuild

atlas <input_filename> -logfile <output_filename>

nanoHUB.org
online simulations and more

Deckbuild Overview

- To start DeckBuild, one needs to type: **deckbuild &**
- When DeckBuild starts, the following application window pops up:

File control buttons
Control room: • commands for defining the problem • switching between simulations • plotting • data optimization
Run-time control buttons
Run-time output window

Used for:

- importing previously saved ASCII files describing a structure of interest
- **Main control** button contains: ***Optimizer*** and ***Examples***

Used for controlling the way the simulator is run:

- **next** - sends current line to simulator
- **run** - runs deck from top to bottom
- **quit** - sends a quit statement to the simulator
- **restart** - restarts the current simulator
- **kill** - kills the simulator

nanoHUB.org
online simulations and more

ATLAS Syntax

- The form of the input file statements is:

  <STATEMENT> <PARAMETER> = <VALUE>

  The parameter can be: real, integer, character and logical.

- The order in which the ATLAS commands occur is the following:

  A) Structure specification: MESH, REGION, ELECTRODE, DOPING

  B) Material models specification: MATERIAL, MODELS, CONTACT, INTERFACE

  C) Numerical method selection: METHOD

  D) Solution specification: LOG, SOLVE, LOAD, SAVE

  E) Results analysis: EXTRACT, TONYPLOT

- The input file can be created using the DeckBuild Command Menu:

  Commands/Command Menu

nanoHUB.org
online simulations and more

(A) Structure Specification

- **MESH** statement specification

  INFILE, OUTFILE ➔ file with previously saved mesh, new file
  SPACE.MULT ➔ scale factor applied to all specified grid spacing
  CYLINDRICAL, RECTANGULAR ➔ describes mesh symmetry
  NX, NY ➔ number of nodes along the *x*- and *y*-direction

  mesh nx=36 ny=30

- **X.MESH, Y.MESH** statements - Specify the location of grid lines along the *x*- and *y*-axes

  NODE ➔ specifies mesh line index
  LOCATION ➔ specifies the location of the grid line
  RATIO ➔ ratio to be used when interpolating grid lines between given locations
  SPACING ➔ specifies mesh spacing at a given location

  x.mesh loc = 0.0 spacing = 0.2
  x.mesh loc = 0.85 spacing = 0.01
  x.mesh loc = 2 spacing = 0.3

nanoHUB.org
online simulations and more

- **ELIMINATE** statement

  Eliminates every second mesh point in a rectangular grid specified by **X.MIN**, **X.MAX**, **Y.MIN** and **Y.MAX**

  COLUMNS, ROWS ➔ columns, rows elimination

  eliminate x.min=0 x.max=4 y.min=0 y.max=3

- **REGION** statement - Specifies regions and materials

  NUMBER ➔ denotes region number

  material ➔ can be **SILICON**, **OXIDE**

  position ➔ defines the location of the region in terms of (1) actual position and (2) grid nodes

  region num=1 ix.lo=1 ix.hi=25 iy.lo=1 iy.hi=20 silicon
  region num=1 y.max=0 oxide
  region num=2 y.min=0 silicon

- **ELECTRODE** statement - must specify at least one electrode within the simulation domain

  NAME - defines the name of the electrode: **SOURCE**, **DRAIN**, **GATE**

  position parameter - **BOTTOM**, **LEFT**, **RIGHT**, **TOP**, **SUBSTRATE**, **IX.LOW**, **IX.HIGH**, **X.MIN**, **X.MAX**, **LENGTH**

nanoHUB.org
online simulations and more

- **DOPING** statement

  Can be used to set the doping profile analytically. Analytical doping profiles can be defined with the following parameters:

  distribution type ➔ **UNIFORM**, **GAUSSIAN**

  doping type ➔ **N.TYPE**, **P.TYPE**

  CONCENTRATION ➔ peak concentration specification for Gaussian profiles

  CHARACTERISTIC ➔ principal characteristic length of the implant (standard deviation). One can specify junction depth instead.

  PEAK ➔ specifies the location of a peak of a Gaussian profile

  position ➔ **X.LEFT**, **X.RIGHT, REGION**

  doping uniform concentration=1E16 n.type region=1
  **doping gaussian concentration=1E18 characteristic=0.05 **
  p.type x.left=0 x.right=1.0 peak=0.1

  The doping profile can also be imported from SSUPREM3. One must use the **MASTER** parameter in the doping statement combined with the **INFILE** parameter to be able to properly import the doping profile.

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

- **COMMENTS ON THE MESH SET-UP**

  (1) Defining a good mesh is a crucial issue in device simulations. There are several factors that need to be taken into account when setting the mesh:

  ACCURACY - fine mesh is needed to properly resolve the structure

  EFFICIENCY - for the simulation to finish in a reasonable time, fewer grid points must be used

  (2) Critical areas where fine mesh is needed include

  depletion regions: high-field regions

  Si/SiO2 interface: high transverse electric field region

  emitter/base junction of a BJT: recombination is important

  impact ionization areas

- **REGRID** statement allows fine mesh generation in critical device areas. This statement is used after the MESH, REGION, MATERIAL, ELECTRODE, and DOPING statements. There are two ways in which regridding can be done:

  regrid on **DOPING**

  regrid using **SOLUTION VARIABLES**

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

(B) Materials Models Specification

- **CONTACT** statement

  NAME ➔ specifies the name of the contact: GATE, DRAIN, ANODE

  WORKFUNCTION ➔ specifies workfunction of a metal, or if specifies N.POLYSILICON, then it implicitly assumes one

  type ➔ specifies the type of a contact: CURRENT, VOLTAGE, FLOATING

  CONTACT IMPEDANCE ➔ uses RESISTANCE, CAPACITANCE, INDUCTANCE, CON.RESISTANCE (used for distributed contact resistance specification)

  Schottky barrier ➔ BARRIER (turns on barrier lowering mechanism), ALPHA (specification of the barrier lowering)

```
contact  name=gate  workfunction=4.8
contact  name=gate  n.polysilicon
contact  name=drain  current
contact  name=drain  resistance=40.0  \
         capacitance=20.E-12  inductance=1.E-6
```

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

- **MATERIAL** statement

  Atlas also supplies a default list of parameters for the properties of the material used in the simulation. The parameters specified in the MATERIAL statement include, for example: *electron affinity*, *energy bandgap*, *density of states function*, *saturation velocities*, *minority carrier lifetimes*, *Auger* and *impact ionization coefficients*, etc.

  REGION ➔ specifies the region number to which the above-described parameters apply

  parameters ➔ Some of the most commonly used parameters include: AFFINITY, EG300, MUN, MUP, NC300, NV300, PERMITTIVITY, TAUN0, TAUP0, VSATN, VSATP

```
material  taun0=5.0E-6  taup0=5.0E-6  mun=3000  \
          mup=500  region=2
material  material=silicon  eg300=1.2  mun=1100
```

- **INTERFACE** statement – Specifies interface charge density and surface recombination velocity.

  QF, S.N, S.P ➔ amount of interface charge density, surface recombination velocity for electrons and holes

```
interface  qf=3E10  x.min=1.  x.max=2.  y.min=0.  y.max=0.5
interface  y.min=0  s.n=1E4  s.p=1E4
```


nanoHUB.org
online simulations and more

- **MODELS** and **IMPACT** statements

  The physical models that are specified with the MODELS and IMPACT statements include:

  mobility model ➔ CONMOB, ANALYTIC, ARORA, FLDMOB, TASCH, etc.
  recombination models ➔ SRH, CONSRH, AUGER, OPTR
  carrier statistics ➔ BOLTZMANN, FERMI, INCOMPLETE, IONIZ, BGN
  impact ionization ➔ CROWELL, SELB
  tunneling model ➔ FNORD, BBT.STD (band to band - direct transitions), BBT.KL (direct and indirect transitions), HEI and HHI (hot electron and hot hole injection)

```
models  conmob  fldmob  srh  fermidirac
impact selb
```

  Additional important parameters that can be specified within the MODELS statement include:

  NUMCARR ➔ specifies number of carriers, and is followed by a carrier type specification (ELECTRONS or HOLES or both)
  MOS, BIPOLAR ➔ standard models used for MOSFET and BIPOLARs

```
models  MOS  numcarr=1  holes
models BIP print
```

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

(C) Numerical Method Selection

- **METHOD** statement – allows for several different chices of numerical method selection. The numerical methods that can be specified within the METHOD statement include

  GUMMEL ➔ De-coupled Gummel scheme which solves the necessary equations sequentially, providing linear convergence. Useful when there is weak coupling between the resultant equations.

  NEWTON ➔ Provides quadratic convergence, and needs to be used for the case of strong coupling between the resultant equations.

  BLOCK NEWTON ➔ more efficient than NEWTON method

  ```
  method gummel block newton
  method carriers=0
  ```

One can also alter the parameters relevant for the numerical solution procedure:

CLIMIT.DD ➔ Specifies minimum value of the concentration to be resolved by the solver.

DVMAX ➔ Maximum potential update per iteration. Default value is 1V.

nanoHUB.org
online simulations and more

(D) Solution Specification

ATLAS allows for four different types of solutions to be calculated: ***DC***, ***AC***, ***small signal*** and ***transient*** solutions. The previously set bias at a given electrode is remembered and does not need to be set again.

❶ DC solution procedures and statements:

➔ A stable DC solution is obtained with the following two-step procedure:

- Find good initial guess by solving equilibrium case (initial guess is found based on the local doping density)

  ```
  solve init
  ```

- Step the voltage on a given electrode for a convergent solution:

  ```
  solve vcollector=2.0
  solve vbase=0.0 vstep=0.05 vfinal=1.0 name=base
  ```

➔ To overcome the problems with poor initial guess, one can use the **TRAP** statement, where **MAXTRAPS** is the maximum allowed number of trials (default value is 4)

```
method trap
solve init
solve vdrain=2.0
```

nanoHUB.org
online simulations and more

➔ To generate a family of curves, use the following set of commands:

```
solve  vgate=1.0  outf=solve_vgate1
solve  vgate=2.0  outf=solve_vgate2
load infile=solve_vgate1  log  outfile=mos_drain_sweep1  \
     solve  name=drain  vdrain=0  vfinal=3.3  vstep=0.3
load infile=solve_vgate2  log  outfile=mos_drain_sweep2  \
     solve  name=drain  vdrain=0  vfinal=3.3  vstep=0.3
```

The log statement is used to save the Id/Vds curve from each gate voltage to separate file.

❷ AC solution procedures and statements:

The AC simulation is simply an extension to the DC simulation procedure. The final result of this analysis is the conductance and capacitance between each pair of electrodes. The two types of simulations are:

- Single frequency solution during a DC Ramp

```
solve  vbase=0.  vstep=0.05  vfinal=1  name=base  AC  freq=1e6
```

- Ramped frequency at a single bias

```
solve  vbase=0.7  ac  freq=1e9  fstep=1e9  nfsteps=10
solve  vbase=0.7  ac  freq=1e6  fstep=2  mult.f  nfsteps=10
```

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

❸ Transient solution procedures and statements:

For transient solutions, one needs to use piecewise-linear, exponential and sinusoidal bias functions. For a linear ramp, one needs to specify the following parameters: **TSTART**, **TSTOP**, **TSTEP** and **RAMPTIME**.

```
solve  vgate=1.0  ramptime=1e-9  tstop=10e-9  tstep=1e-11
```

❹ Advanced solution procedures:

- Obtaining solutions around a breakdown point – uses MAXTRAPS
- Using current boundary conditions

  Instead of voltage, one can also specify current boundary conditions. This is important, for example, when simulating BJTs:

```
solve  ibase=1e-6
solve  ibase=1e-6  istep=1e-6  ifinal=5e-6  name=base
```

- The compliance parameter

  This parameter is used to stop simulation when appropriate current level is reached.

```
solve  vgate=1.0
solve  name=drain  vdrain=0  vfinal=2  vstep=0.2  \
       compl=1e-6  cname=drain
```

- The curve trace capability – enables tracing out of complex IV curves

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

(E) Results Analysis

Three types of outputs are produced by the ATLAS tool: *run-time outputs*, *log files* and *solution files*.

❶ Run-time outputs:

The various parameters displayed during the SOLVE statement are listed below:

proj ➔ initial guess methodology used (previous, local or init)

i, ***j***, ***m*** ➔ iteration numbers of the solution and the solution method
- ***i*** = outer loop iteration number
- ***j*** = inner loop number for decoupled solutions
- ***m*** = solution method used: G=Gummel, B=Block, N=Newton

x, ***rhs*** ➔ norms of the equations being solved

(*) ➔ the error measure has met its tolerance

❷ Log files:

The **LOG** parameter is used to store the device characteristics calculated using ATLAS:

```
log  outfile=<file_name>
```


nanoHUB.org
online simulations and more

❸ Solution files:

The syntax to produce the solution files that can be used in conjunction with TonyPlot is:

```
save  outfile=<file_name>
solve  . . . .  outfile=<file_name>.sta   master   [onefileonly]
```

❹ Invoking TonyPlot

➔ To create overlayed plots with TonyPlot, one needs to use the following command:

```
tonyplot  -overlay  file1.log  file2.log
```

➔ To load structure files, containing mesh, doping profile information, etc., one can use the following statement:

```
tonyplot  file.str  -set  mx.set  iv.data
```

This command allows loading of the file called " *file.str* " and sets its display to a previous setup stored in the " *mx.set* " file, and then loads the file containing the *IV*-data.

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

The parameters extraction can be accomplished in two different ways:

1) Using the **EXTRACT** command that operates on previously solved curve or structure file:

➔ To override the default of using open log file, the name of the file that needs to be used is specified in the following manner:

extract init infile="<file_name>"

➔ Parameters that can be extracted using this EXTRACT statement include: threshold voltage, cutoff frequency, etc. The extraction of the threshold voltage is accomplished with the following statement:

extract name="nvt" xintercept(maxslope(curve (v."gate", \ (i."drain"))) -(ave(v."drain"))/2.0)

➔ Default file for saving results is results.final . The results can be stored in other file using the following options:

extract … . Datafile="<file_name>"

2) Using the Functions Menu in TonyPlot that allows one to use saved data for post-computation

3) Using the LOG statement for AC parameter extraction

nanoHUB.org
online simulations and more

Atlas Extract Description

(1) The extract statement can be used in conjunction with:

- **Process extraction**, after running Silvaco ATHENA simulator
- **Device extraction**, after obtaining the electrical characteristics of the device structure being simulated

  - **Log-files: contain the electrical information, more precisely, the *IV*-data obtained via the ATLAS simulation process**
  - **Structure files: contain the additional electrical information, such as electric field, electrostatic potential, etc.**

(2) One can construct a *curve* using separate X and Y-axes. For each of the electrodes, one can choose one of the following: Voltage (*v*), Current (*i*), Capacitance (*c*), Conductance (*g*), Transient time for AC simulations (*time*), Frequency for AC simulations (*frequency*), Temperature (*temperature*), etc.

nanoHUB.org
online simulations and more

(3) More in-depth description of the use of the EXTRACT statement:

- ***Curve***, basic element in the extract statement. The syntax is as follows:

  extract name="curve_name" curve(v."name", i."name")

  "*curve_name*" = name of the curve to which one can refer to in later post-processing steps

- **Axes manipulation:**
  - algebra with a constant (multiplication, division)
  - operators application (*abs*, *log*, *log10*, *sqrt*)

- **Curve manipulation primitives:**

  min, *max*, *ave*, *minslope*, *maxslope*, *slope*, *xintercept*, *yintercept*, *x.val* from curve where *y.val=Y* (*val.occno=1*, would mean first occurrence of the preset condition)

- **Example:** Find max $\beta = I_C/I_B$ vs. I_C

  extract "maxbeta" max(curve(i."colector", i."colector"/i."base"))

(*) Additional set of examples for the EXTRACT statement can be found in the Silvaco ATLAS manual: *VWF Interactive Tools – part I*

Network for Computational Nanotechnology

nanoHUB.org
online simulations and more

Diode Example

```
go atlas

#   MESH SPECIFICATION PART
    mesh  space.mult=1.0
#
    x.mesh loc=0.00 spac=0.5
    x.mesh loc=3.00 spac=0.2
    x.mesh loc=5.00 spac=0.25
    x.mesh loc=7.00 spac=0.25
    x.mesh loc=9.00 spac=0.2
    x.mesh loc=12.00 spac=0.5
#
    y.mesh loc=0.00 spac=0.1
    y.mesh loc=1.00 spac=0.1
    y.mesh loc=2.00 spac=0.2
    y.mesh loc=5.00 spac=0.4

#   REGIONS AND ELECTRODES SPECIFICATION
    region  num=1  silicon

    electr  name=anode  x.min=5  length=2
    electr  name=cathode  bot
```

Network for Computational Nanotechnology

nanoHUB.org

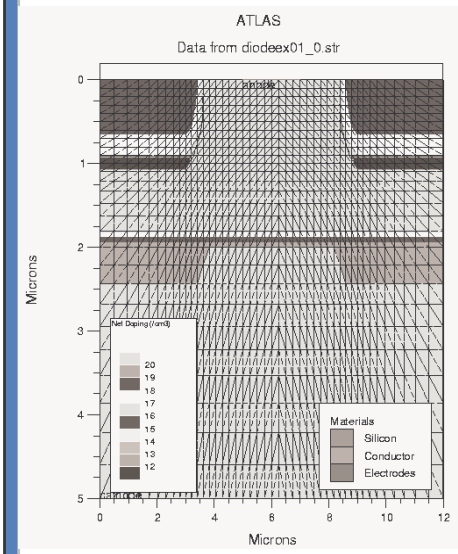
online simulations and more

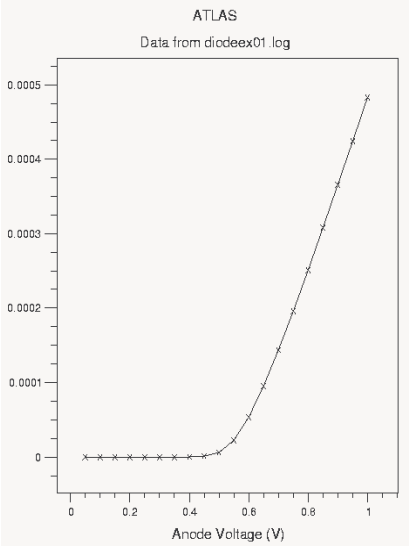
```
#     DOPING SPECIFICATION
#....  N-epi doping
doping  n.type conc=5.e16 uniform

#....  Guardring doping
doping   p.type conc=1e19 x.min=0  x.max=3  junc=1 rat=0.6 gauss
doping   p.type conc=1e19 x.min=9 x.max=12 junc=1 rat=0.6 gauss

#....  N+ doping
doping  n.type conc=1e20 x.min=0 x.max=12 y.top=2 y.bottom=5 uniform

#     SAVING THE MESH
save outf=diodeex01_0.str
tonyplot diodeex01_0.str -set diodeex01_0.set

#     MODELS SPECIFICATION
model    conmob  fldmob  srh  auger  bgn
contact    name=anode workf=4.97

#     SOLUTION PART
#.... Initial solution part
solve      init
method newton

#.... Stepping the anode voltage and saving the data
log   outfile=diodeex01.log
Solve  vanode=0.05  vstep=0.05  vfinal=1  name=anode
tonyplot diodeex01.log -set diodeex01_log.set
quit
```

Network for Computational Nanotechnology

nanoHUB.org

online simulations and more





Network for Computational Nanotechnology