

VAPP: Instructions for Hands-On Examples

1. Setup MAPP + VAPP:

By now you should have a working MAPP + VAPP setup on your machine. If not, get one of us (Archit, Tianshi, Gokcen or Jaijeet) to help you.

2. Run a Full-Blown Test Script:

At a MATLAB prompt: run the FEFET test script by typing the following:

Change into workshop directory:	cd workshop
Change into Alam_FEFET directory:	cd Alam_FEFET
Translate the FEFET model:	va2modspec('mvs_5t_mod2.va')
Calculate characteristic curves with MAPP:	run_char_curves_Alam_fefet_combined

3. Go back to basics – Create your own VA model and import it into MAPP using VAPP

Go to the “**diode**” directory under the workshop sub-directory in your VAPP installation and follow these steps in order to implement a simple diode Model in Verilog-A and import in into VAPP.

I. Simple diode

- Open the “**diode.va**” file and find

```
I(br_dio) <+
```

statement on line 16.

Complete the missing RHS of the contribution statement with

```
Is * (exp((V(br_dio))/ $vt(Temp)) - 1);
```

- Import the diode model into MAPP

```
va2modspec('diode.va');
```

- Compute the IV-curve of the diode by running

```
run_char_curves_diode;
```

The script will ask you to connect sources to the terminals of the device.

Choose V or I for terminal p. [V]:

type V ↵

Choose V or I for terminal n. [V]:

type V ↵

You should see the I-V curve of the diode now.

- Now edit the file “run_char_curves_diode.m”
Change the voltage value in line 5 from 0.7 to 1

```
Vhigh = 1;
```

- Run the IV-curve calculation again:

```
run_char_curves_diode;
```

- MAPP should quit and display an error.
What is the reason?

II. Resistor + Diode

- Add an internal node, int, to your model at line 7

```
electrical p, int, n;
```

- Connect the second node of the br_dio branch to the internal node
and add a second branch, br_res

```
branch(p, int) br_dio;  
branch(int, n) br_res;
```

- Add the resistor equation to your model

```
I(res) <+ V(res)/R;
```

- Import the new diode model into MAPP

```
va2modspec('diode.va');
```

- Try running the transient analysis with the new model

```
run_char_curves_diode;
```

4. Ferro-electric FET

A ferro-electric FET is a transistor with a gate capacitance that has negative capacitance regions.

To investigate this behavior:

- Cd into the `Alam_FEFET` directory
- Translate the two models

```
va2modspec('mvs_5t_mod.va');
va2modspec('neg_cap_3t.va');
```

- Compute the non-linear charge-voltage characteristic of the negative capacitance device

```
run_dc_sweep_neg_cap_3t;
```

- Run the homotopy analysis

```
run_char_curves_Alam_fefet;
```

- Run the analysis on the simple inverter circuit involving a FEFET

```
run_inverter_homotopy_Alam_fefet;
```

5. Bsim3 Ring Oscillator

Go to the `bsim3_ringosc` directory and translate the BSIM model. After that you can run the `run_transient_bsim3_ringosc` script to perform a transient simulation and see the oscillatory behavior of the circuit.

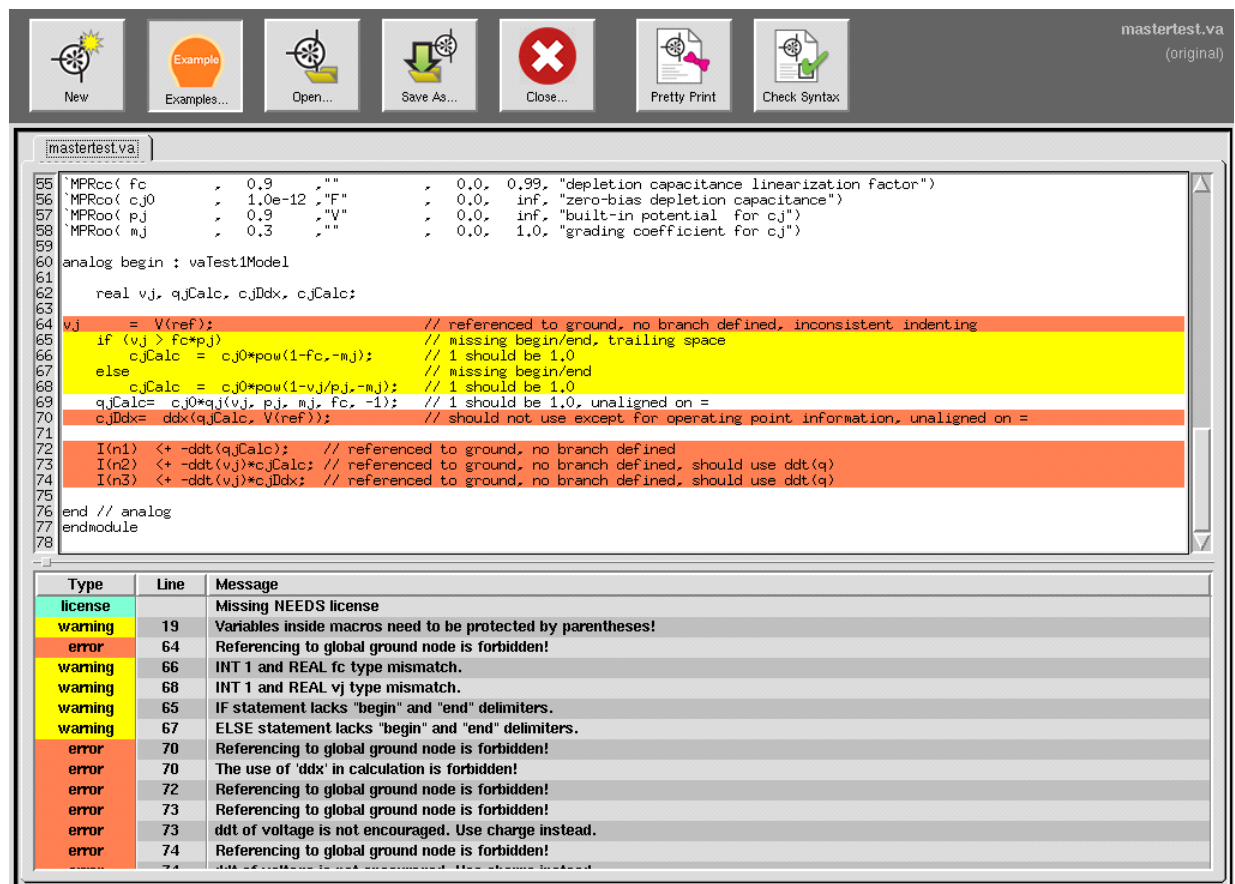
This can take a long time. You can explore the model and the simulation script while we wait.

6. VALint

At any time during the workshop, you may access VALint tool at

<https://nanohub.org/resources/vachecker>.

VALint is the NEEDS created, automatic Verilog-A code checker to check the quality of the Verilog-A code and provide the author feedback if bad practices, common mistakes, pitfalls, or inefficiencies are found.



The screenshot displays the VALint tool interface. At the top, there is a toolbar with icons for New, Examples..., Open..., Save As..., Close..., Pretty Print, and Check Syntax. The main window shows a Verilog-A code file named 'masterTest.va'. The code includes parameter definitions for `fc`, `cj0`, `pj`, and `mj`, followed by an `analog` block containing several assignments and conditional statements. Several lines of code are highlighted with orange and yellow backgrounds, indicating errors or warnings. Below the code editor, a table lists the detected errors and warnings.

Type	Line	Message
license		Missing NEEDS license
warning	19	Variables inside macros need to be protected by parentheses!
error	64	Referencing to global ground node is forbidden!
warning	66	INT 1 and REAL fc type mismatch.
warning	68	INT 1 and REAL vj type mismatch.
warning	65	IF statement lacks "begin" and "end" delimiters.
warning	67	ELSE statement lacks "begin" and "end" delimiters.
error	70	Referencing to global ground node is forbidden!
error	70	The use of 'ddx' in calculation is forbidden!
error	72	Referencing to global ground node is forbidden!
error	73	Referencing to global ground node is forbidden!
error	73	ddt of voltage is not encouraged. Use charge instead.
error	74	Referencing to global ground node is forbidden!