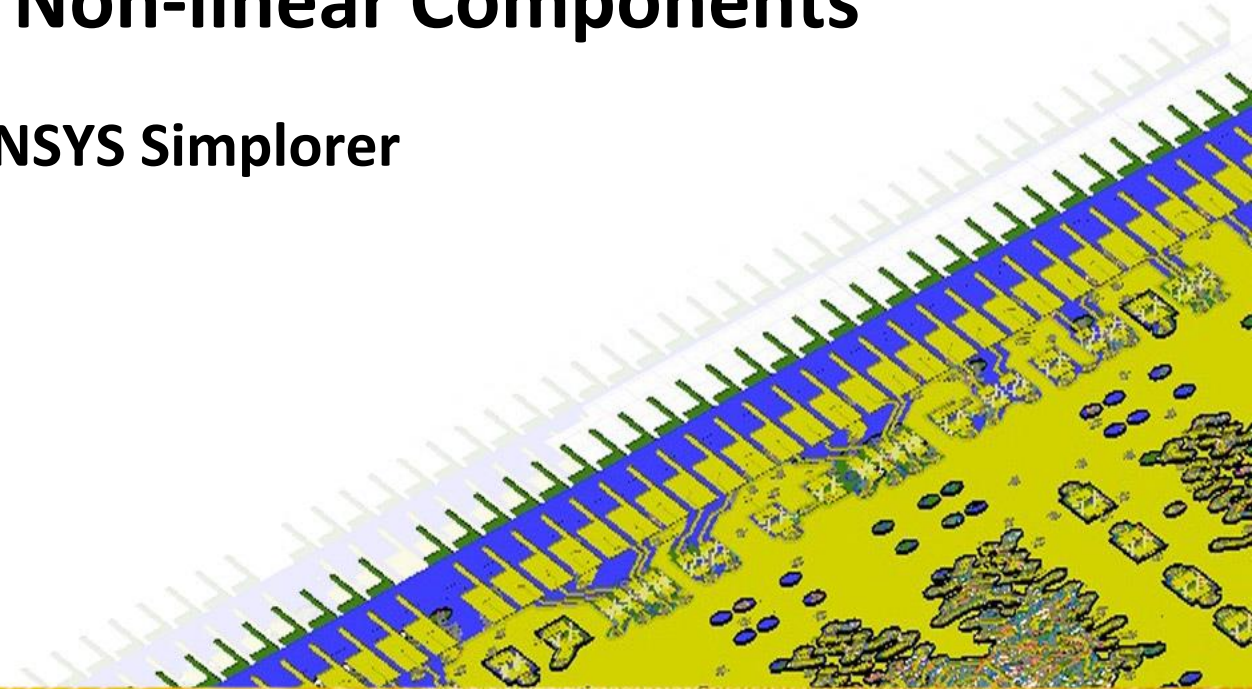




Module 04: VHDL-AMS, Library Management, Device Characterization, Non-linear Components

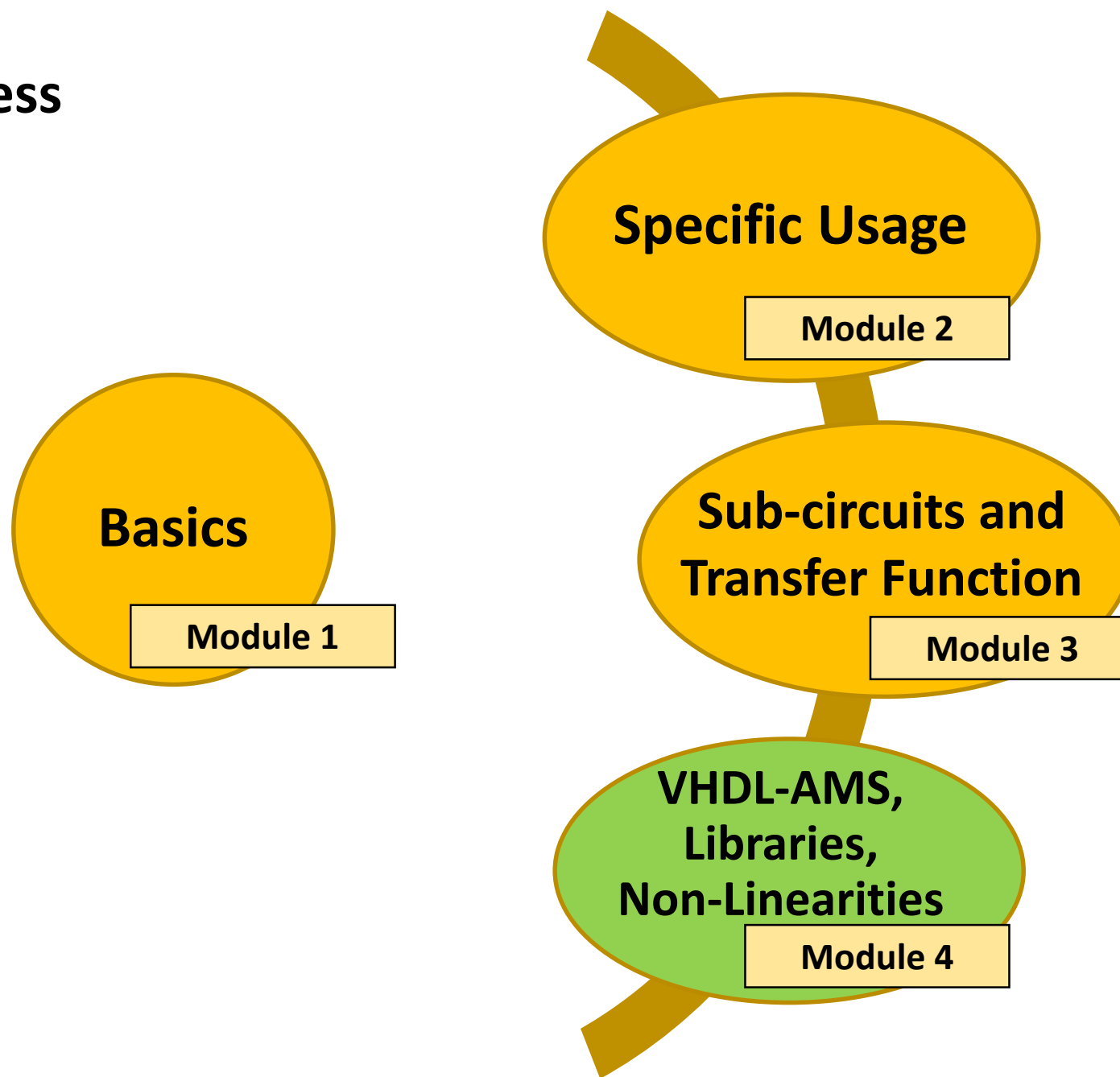
Introduction to ANSYS Simplorer



Overview

- **VHDL-AMS**
 - Language
 - Entity and Architecture(s)
- **Library Management**
- **Device Characterization**
 - Sheetscan
 - Characterization Workflow
- **Non-Linear Components**
- **Workshop 4.1: VHDL component and export to library**
- **Workshop 4.2: Non-linear Components**

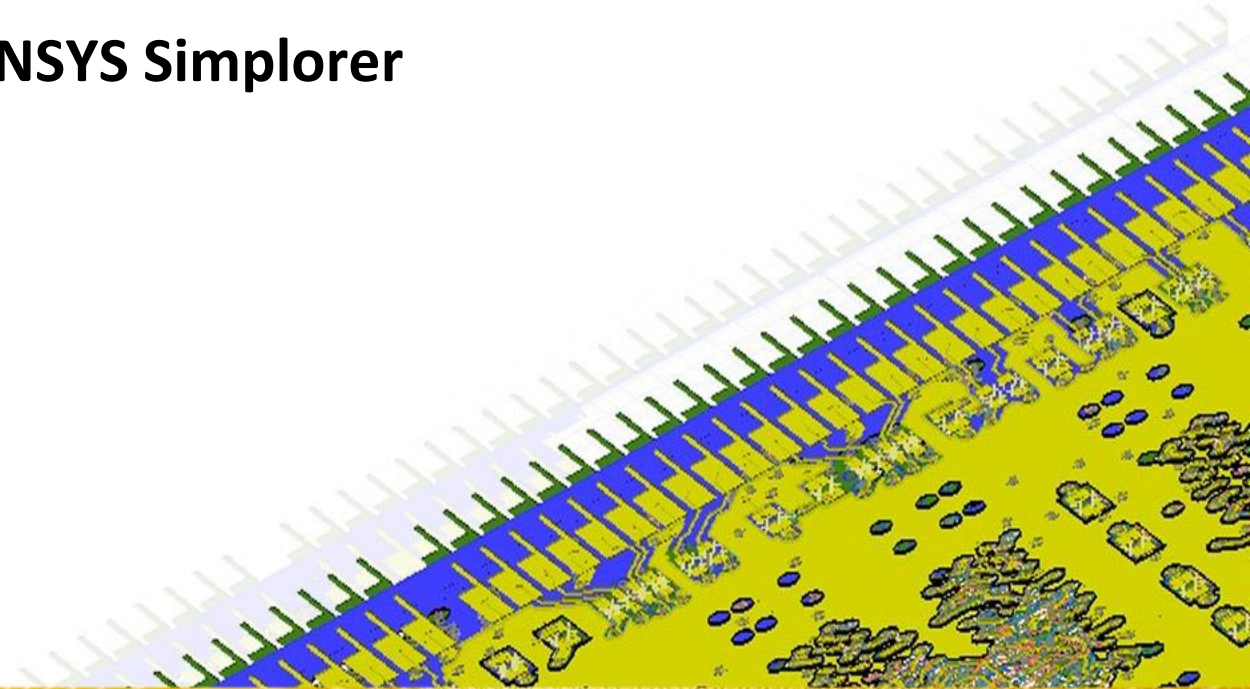
Overall Process





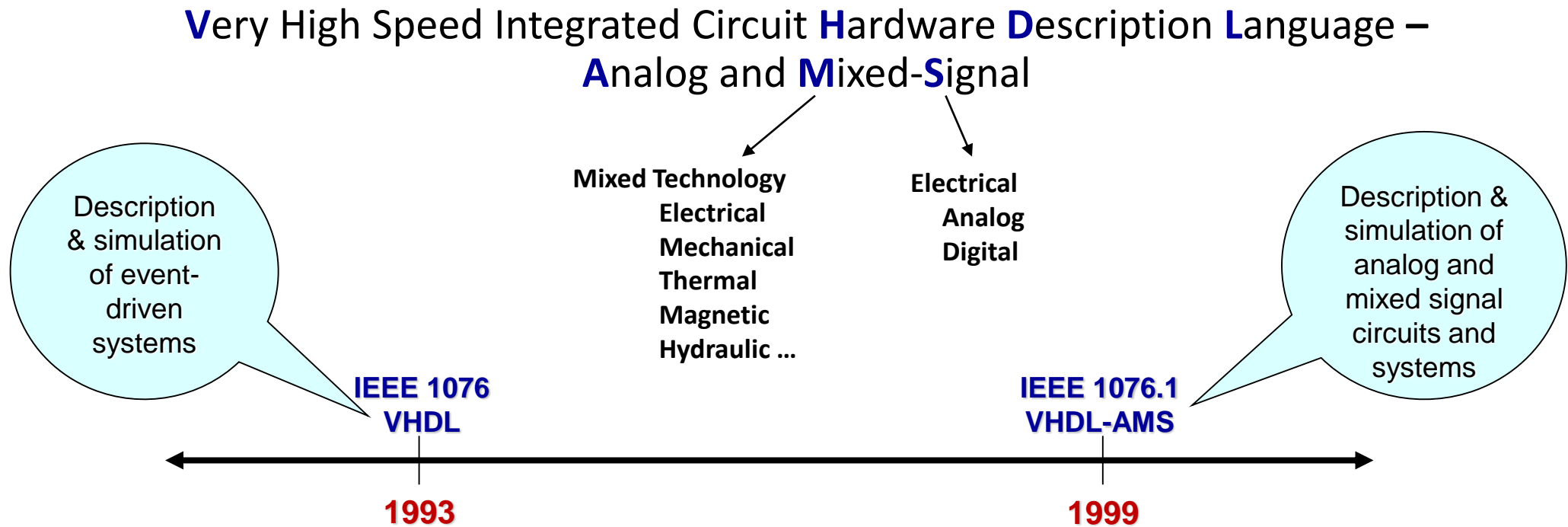
VHDL-AMS

Introduction to ANSYS Simplorer



VHDL-AMS (Modeling Language)

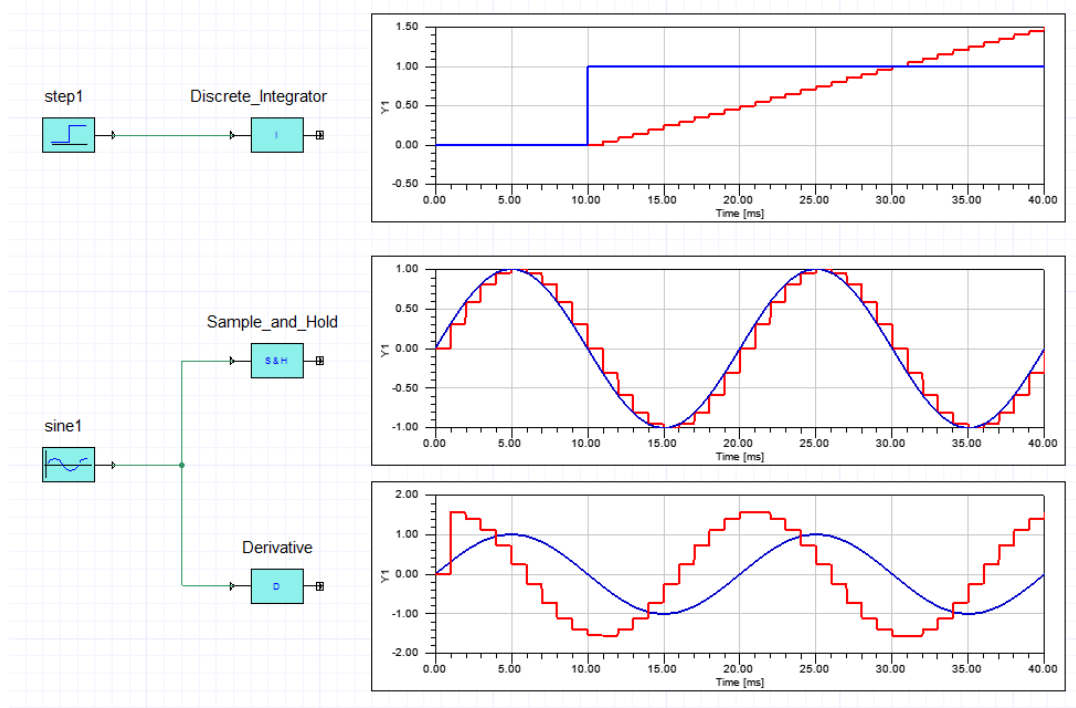
- Simplorer offers users different possibilities to create their own customized models
- One of those possibilities is the VHDL-AMS Programming language
- Born in 1993, VHDL was originally created as a standardized language only for digital components
- In 1999 VHDL was expanded to VHDL-AMS, allowing programming analog objects of different Nature



VHDL-AMS is a superset of VHDL IEEE Std. 1076

Advantages of VHDL-AMS

- Main advantages of VHDL-AMS are flexibility and compatibility
- Standard Format Allows Model Portability
 - Between different engineering groups within same company
 - With Sub-Contractors
 - Between different simulators
- Multi-level Modeling
 - Different levels of abstraction of model behavior
- Multi-domain Modeling
 - Electrical, Thermal, Magnetic, Mechanical, etc.
- Mixed-signal Modeling
 - Supports analog and digital modeling



VHDL-AMS Basic Structure

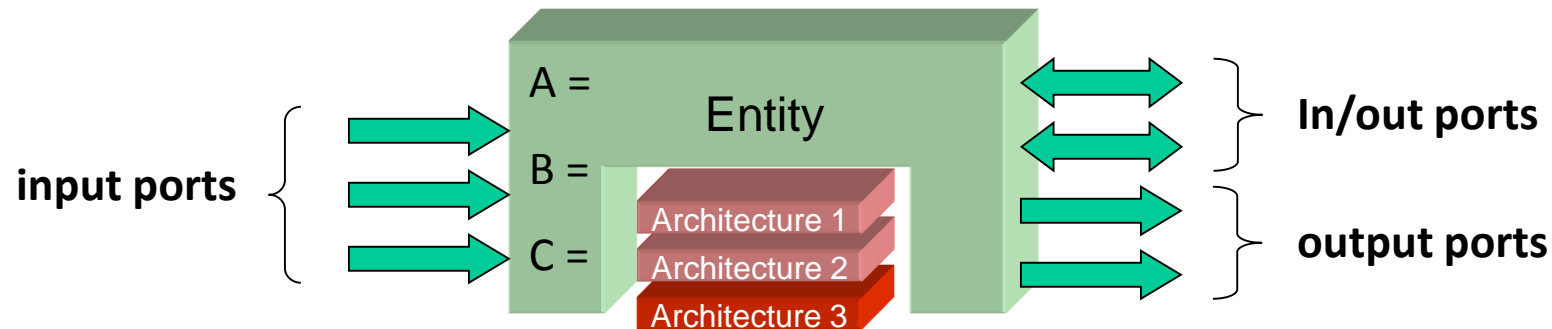
- Every component created in VHDL-AMS presents a structure divided into two sub-categories: **Entity** and **Architecture(s)**

- **Entity**

- Interface description of a subsystem or physical device
- Specifies input and output ports to the model
- Can specify input parameters to the model (A =, B =, C = ..)

- **Architecture(s)**

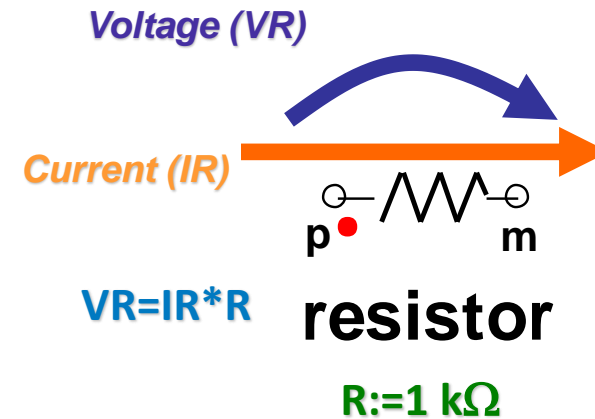
- Can provide several levels of model description
- Modeling can deal with both analog (continuous) and digital (discrete) domains



VHDL-AMS Simple Example

```
ENTITY resistor IS
  PORT (QUANTITY R : RESISTANCE := 1.0e+03);
        (TERMINAL p,m : ELECTRICAL);
END ENTITY resistor;
```

```
ARCHITECTURE behav OF resistor IS
  QUANTITY VR ACROSS IR THROUGH p TO m;
BEGIN
  VR == IR * R;
END ARCHITECTURE behav;
```



RLC Component (Multiple Architectures)

```
LIBRARY IEEE;  
USE IEEE.ELECTRICAL_SYSTEMS.ALL;  
ENTITY my_component IS  
    GENERIC (value : REAL := 1.0e+3);  
    PORT (TERMINAL p,m : ELECTRICAL);  
END ENTITY my_component;
```

```
ARCHITECTURE resistor OF my_component IS  
    QUANTITY voltage ACROSS current THROUGH p TO m;  
BEGIN  
    voltage == current * value;  
END ARCHITECTURE resistor;
```

$$VR = I * R$$

```
ARCHITECTURE capacitor OF my_component IS  
    QUANTITY voltage ACROSS current THROUGH p TO m;  
BEGIN  
    current == value * voltage'DOT;  
END ARCHITECTURE capacitor;
```

$$iC = C * dv/dt$$

```
ARCHITECTURE inductor OF my_component IS  
    QUANTITY voltage ACROSS current THROUGH p TO m;  
BEGIN  
    voltage == value * current'DOT;  
END ARCHITECTURE inductor;
```

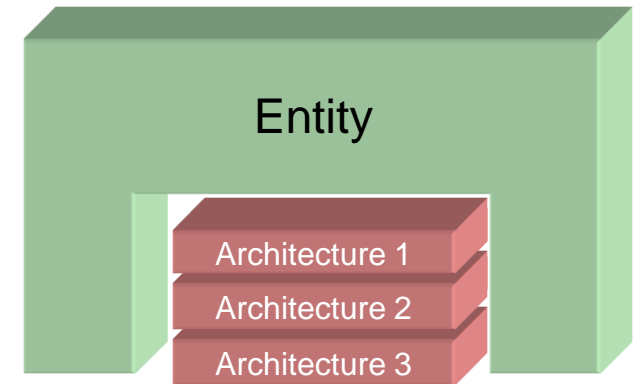
$$VL = L * di/dt$$

Entity

Arch. 1

Arch. 2

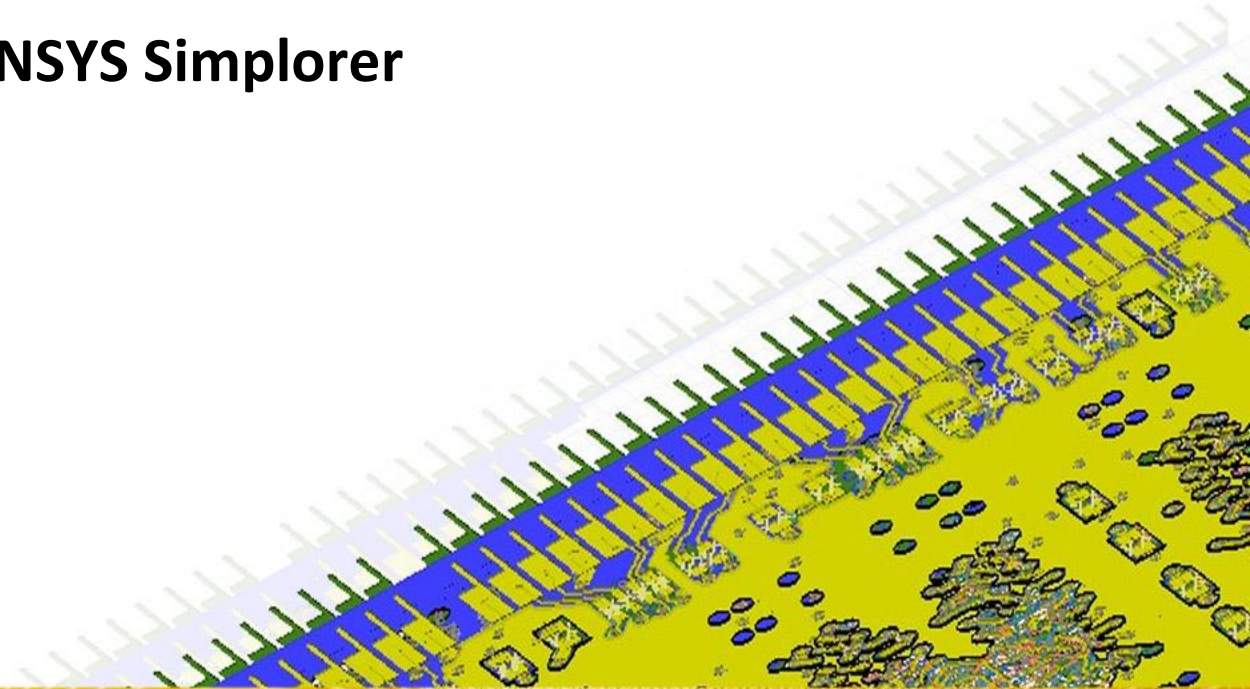
Arch. 3





Library Management

Introduction to ANSYS Simplorer



Semiconductor device modeling

- Different levels of abstraction

- Ideal switch

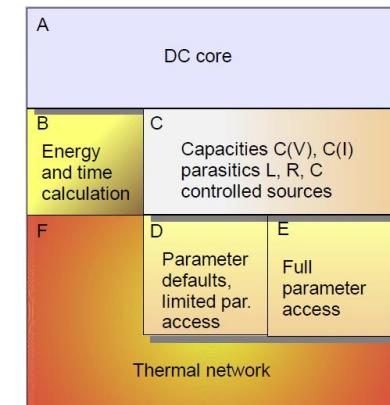
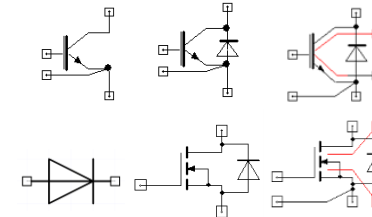
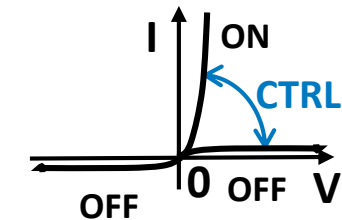
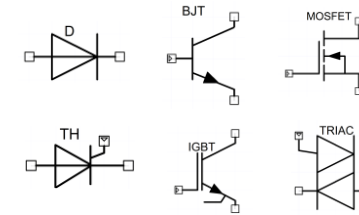
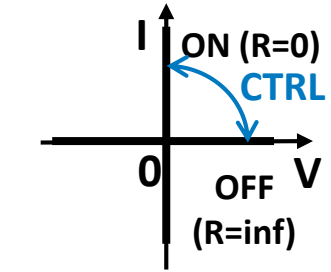
- A logical signal controls the state of the switch
 - The resistance of the line connection is zero

- Semiconductor System Level

- Simulate a static voltage-current-relation
$$v(t) = f(i(t), [CTRL])$$
 - To control device, apply a logical signal

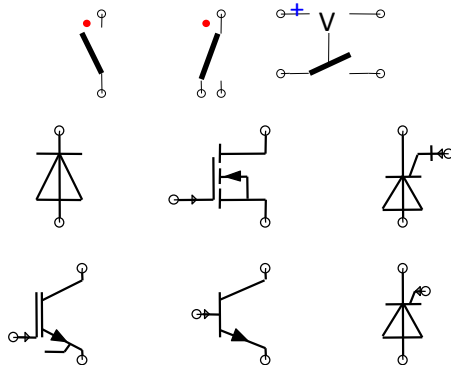
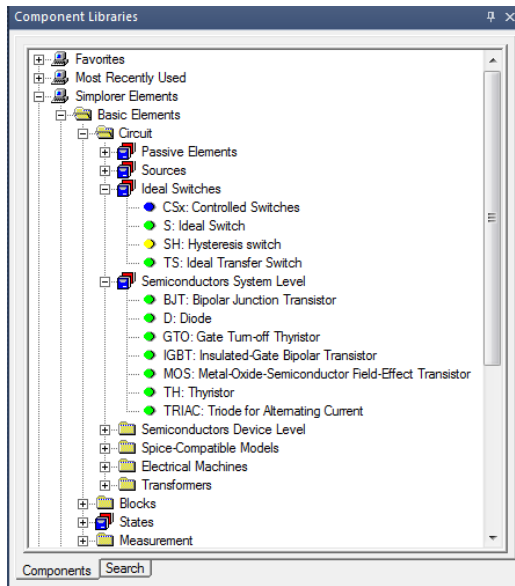
- Semiconductor Device Level

- Modular models with definable simulation levels
 - Represent temperature dependent static and dynamic behavior of semiconductor devices
 - Can include thermal network
 - To control a switch electrical driver circuit is needed
 - Can be parametrized using Semiconductor Device Characterization Wizard

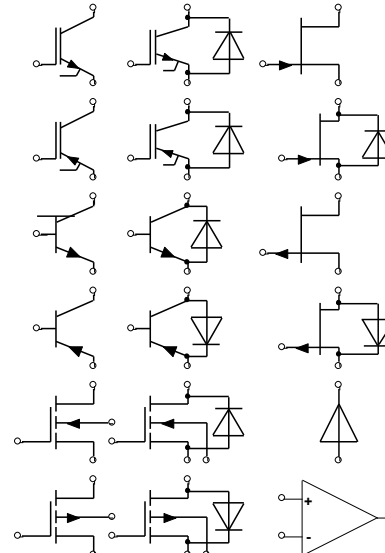
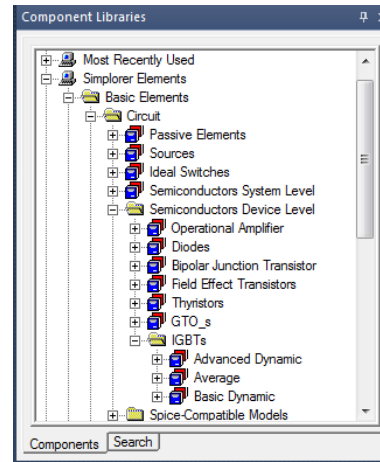


Library - Semiconductors

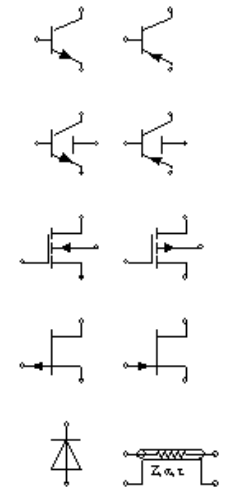
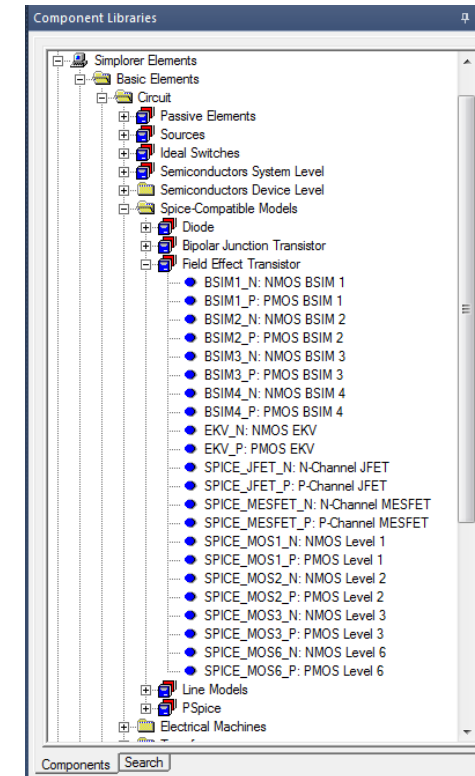
Ideal switches and semiconductor - system level



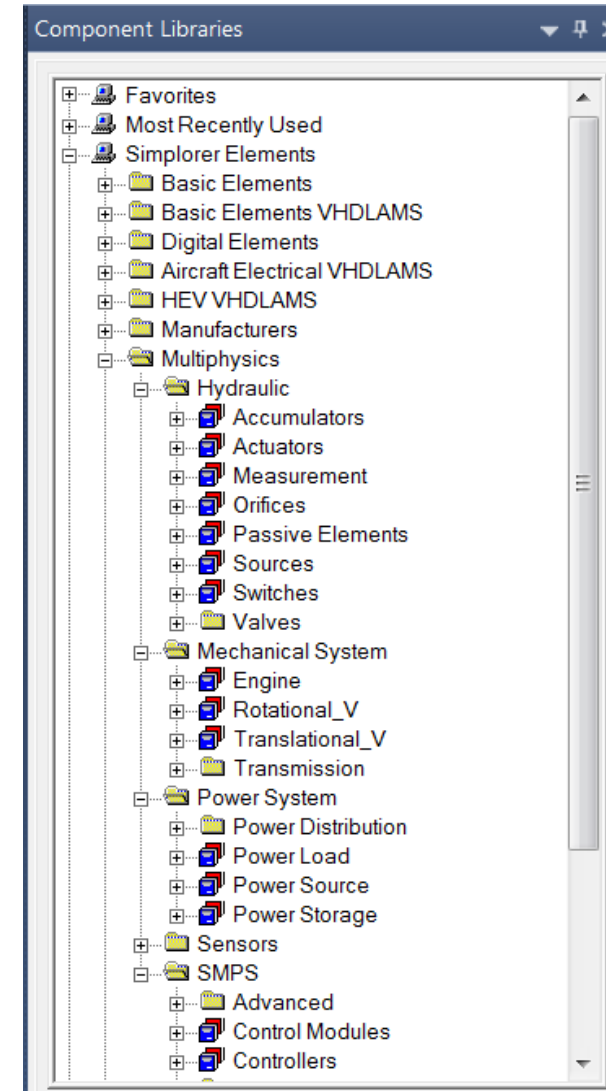
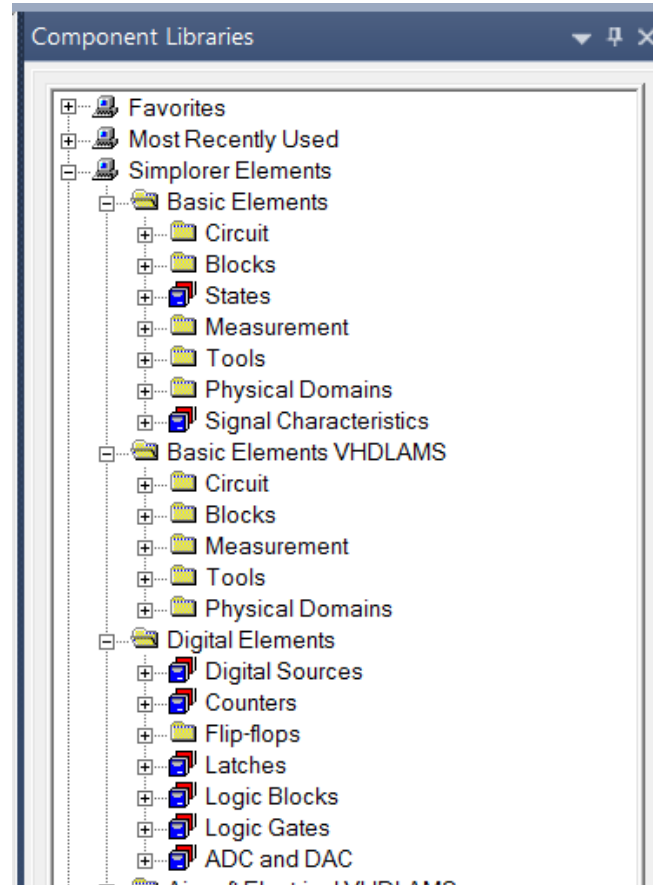
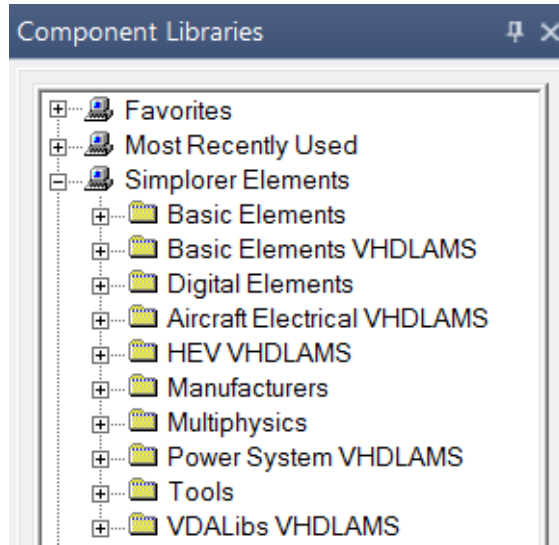
Semiconductor - device level



Spice compatible models

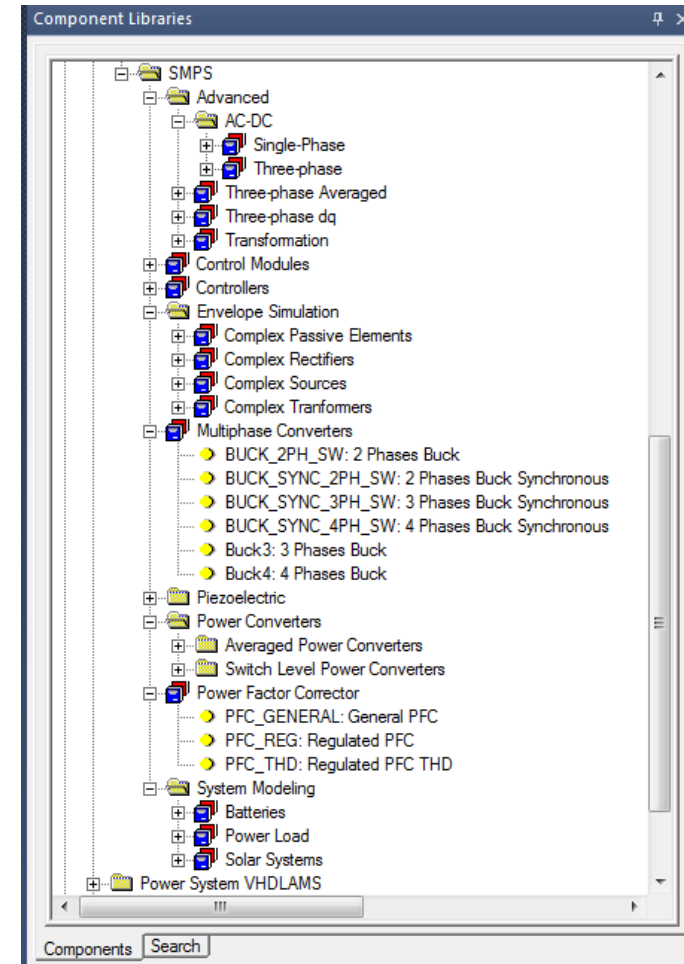


Libraries Examples



SMPS Library

- Switched Mode Power Supply (SMPS) library provides a comprehensive set of predefined power electronic circuit topologies and related control algorithms for the design of power converters
- SMPS library is under *Simplorer Elements* → *Multiphysics*
- Graphical modeling
- Predefined models for frequently used DC/DC converter topologies (including PFC, isolated and non-isolated designs)
- AC/DC converters like 1-Ph and 3-Ph rectifiers, system energy storage elements, loads and sources aid in the overall power system design
- Common power electronic control algorithms and control blocks, such as PWMs, PID, $abc \leftrightarrow dq$, etc.
- Average models can be used for control loop design, AC analysis, and system level converter modeling.



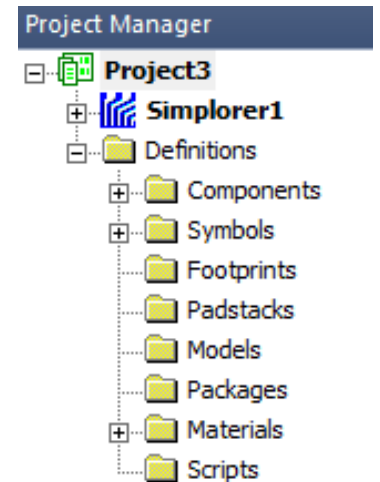
Library Management Introduction

- Component / Definition Concepts

- A component (i.e. a resistor, transistor, motor, etc) needs to have several “definitions” associate with it:
 - “Symbol” definition to describe the graphic appearance
 - “Model” definition to define the characteristic behavior of the component
 - “Component” definition as a high level container
 - Note there are also other types of definitions depending on the component’s need such as “Material” definitions, or if using VHDL-AMS components, sometimes a “Package” definition can be used.

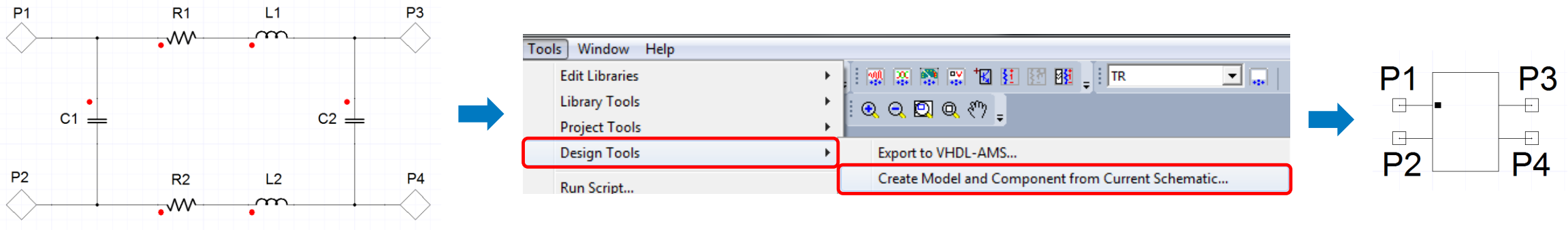
- Library Directories

- There are System library files shipped with Simplorer. These libraries are intended to be read-only (“SysLib”)
- In addition to the System libraries, Simplorer recognizes two user-configurable libraries, “UserLib”, and “PersonalLib”: *Tools → Options → General Options → (General → Directories)* for library paths



Exporting Sub-Circuit to Library

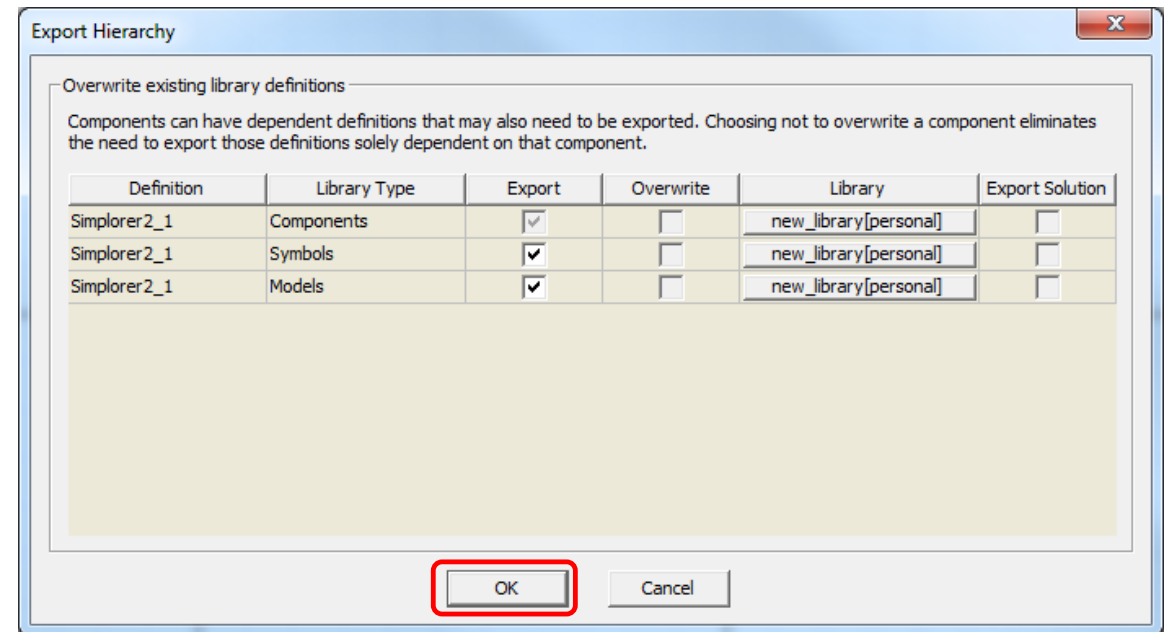
- Simplorer can export an already created sub-circuit to library and save it to a user or personal library, so that it is possible to re-use it in the future. In the following we will assume that a sub-circuit (represented in the picture below) has been already created and this is our starting point
- Note an alternate method to create a sub-circuit can be accessed through menu item **Tools → Design Tools → Create Model and Component from Current Schematic** level



- To add a sub-circuit to a library, select it under the Definitions/Component folder in the Project Manager window, then **RMB → Export to library**

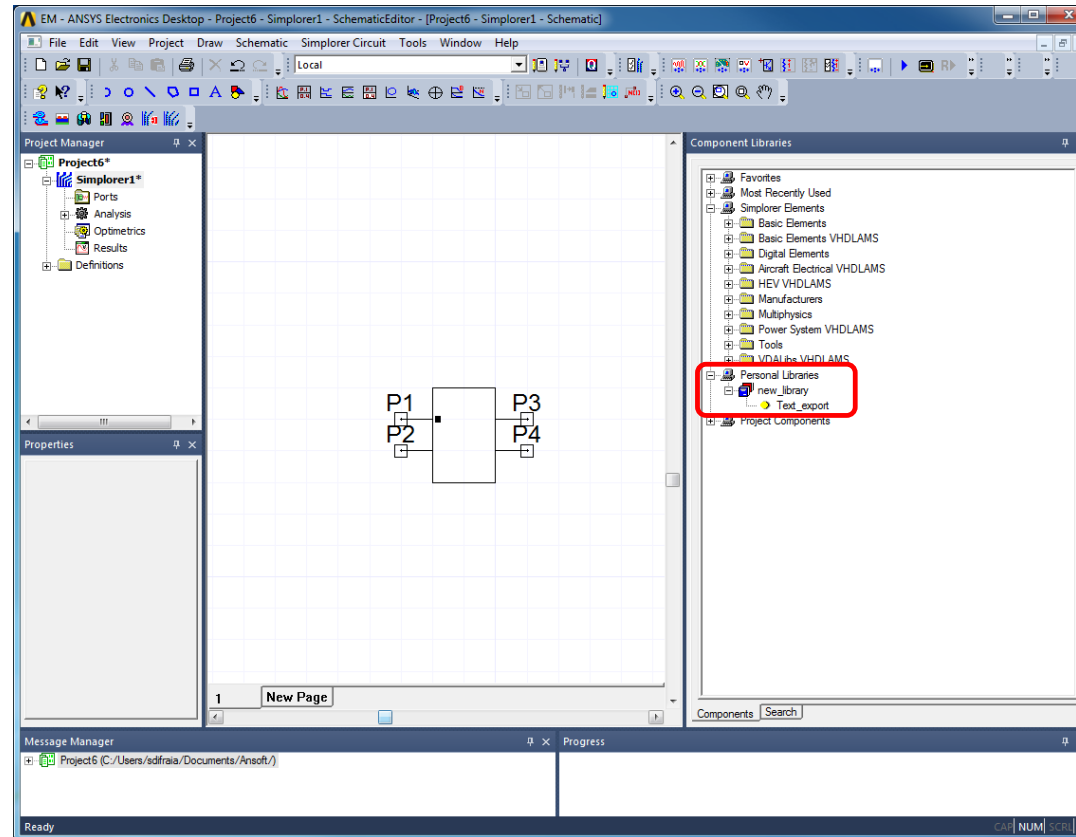
Exporting Sub-Circuit to Library

- A second window appears, allowing users to choose where to save the component (**UserLib** or **PersonalLib**) and the new library name
- Once the Save button is pressed, the Export Hierarchy window pops-up, confirming the export of libraries for **Component**, **Symbol** and **Model**



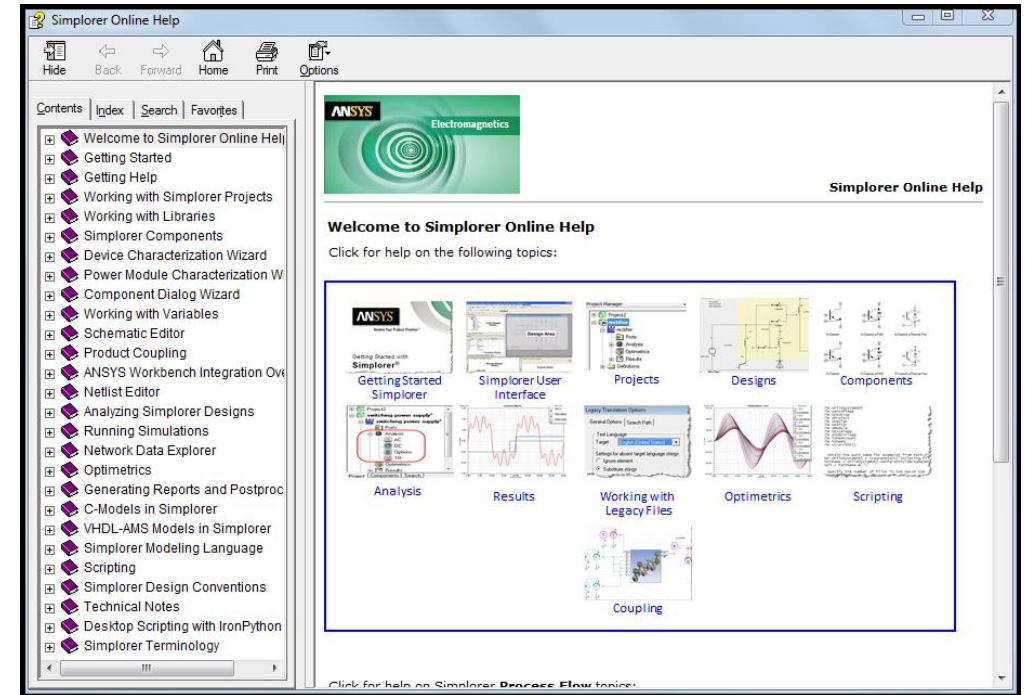
Exporting Sub-Circuit to Library

- In **Component Libraries** window, under **Personal Libraries** appears now the just created **new_library**, containing the **Text_export** component
- In the example below the **Text_export** component, consisting in 4 interface conservative pins, has been placed in the Schematic (the number of pins depends on the complexity of exported component)



Additional Resources

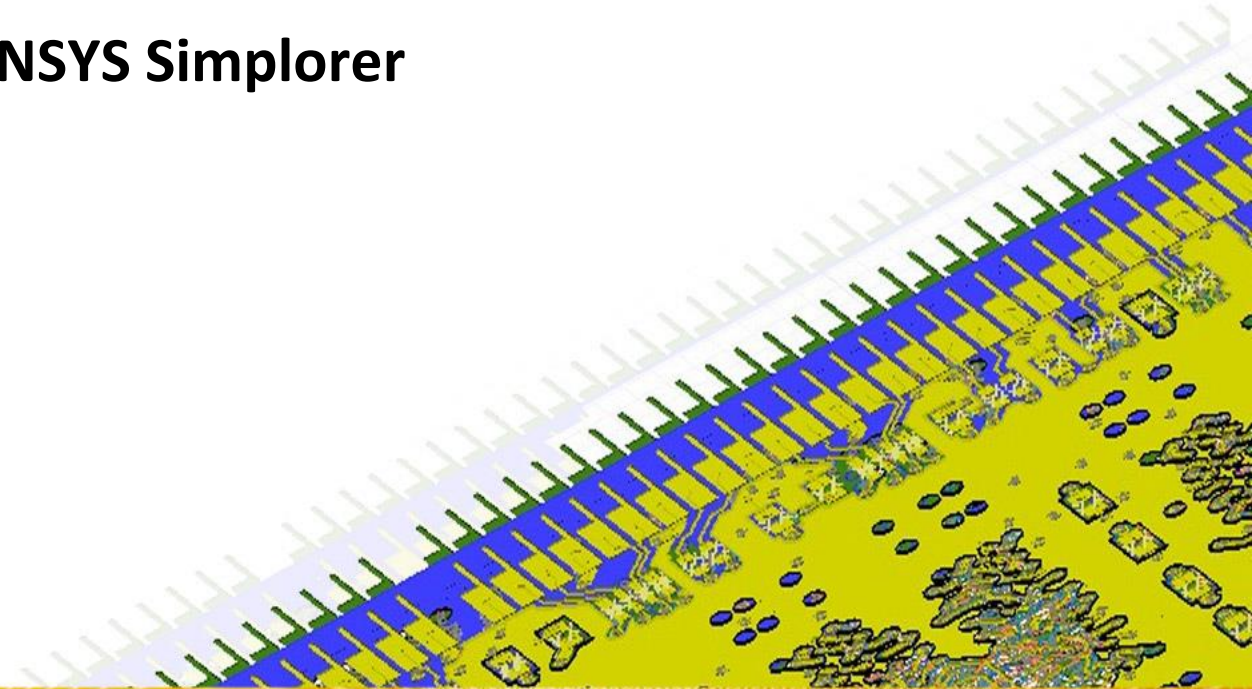
- **Under the AnsysEM Install directory**
 - Program Files\AnsysEM\AnsysEM17.0\Win64
 - Examples directory (Large number of examples)
 - Help/Simplorer directory
 - simplorer.pdf
- **ANSYS Customer Portal**
 - <http://www.ANSYS.com> (select Customer Portal)
 - Latest Patches, Tutorials and Training material, app notes, model libraries
- **Online Help (launch from tools)**
- **Library components have examples/help**





Semiconductor Device Characterization

Introduction to ANSYS Simplorer



Semiconductor Device Characterization

- **Concept**

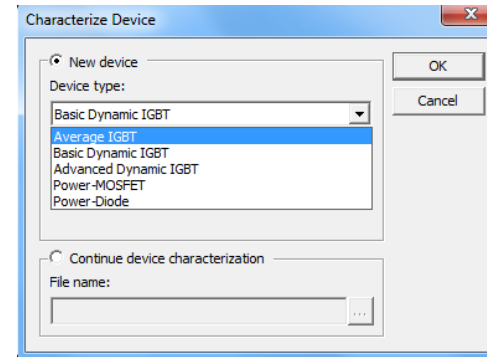
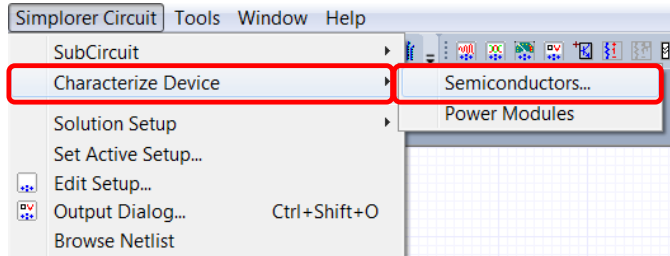
- Simplorer offers the user the possibility to characterize power electronic devices (IGBTs, MOSFETs, Power Diodes) directly from vendor datasheets through a guided process
- Three different levels of accuracy are present:
 - “**Average**” level includes the static characteristics and conducting and switching power calculation for Thermal dynamics
 - “**Basic Dynamic**” level represents in addition reasonably well the rise time, V/I overshoot and some capacitance default effects
 - “**Advanced Dynamic**” level is the most accurate and takes into account all the effects that are possible to be simulated
- The Table in the next slide summarizes the three level of accuracy capabilities

Semiconductor Device Characterization (IGBT, MOSFET, Diode)

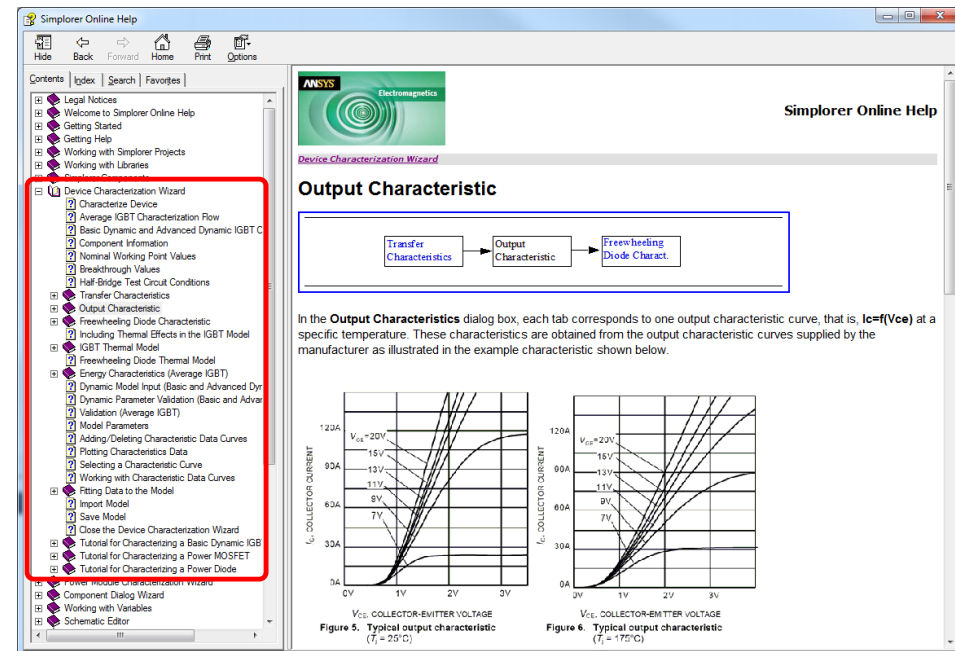
Objective	Average	Basic Dynamic	Advanced Dynamic
DC characteristics	<ul style="list-style-type: none">- Transfer characteristic $I_c(V_{ge})$ accurate- Output characteristic $I_c(V_{ce})$ accurate in the regions of voltage and current saturation- Intrinsic temperature dependency		
Rise time, V/I overshoot	-	Reasonable	More Accurate
Thermal Dynamics	Included		
Gate drive optimization	-	Initial design	More Suitable
Capacitance Models	-	Default C(V)	Full access to the C(V) characteristics

Semiconductor Device Characterization

- The Semiconductor Device Characterization Wizard can be launched from menu item *Simplorer Circuit* → *Characterize Device* → *Semiconductors*



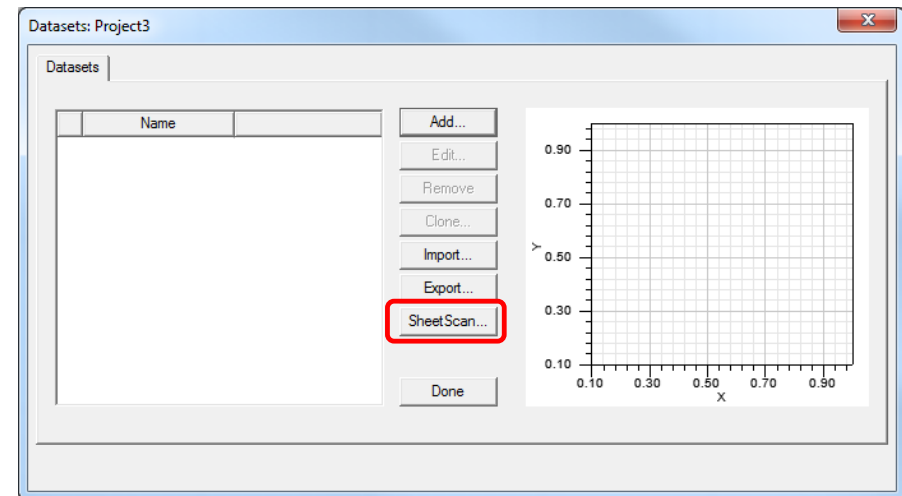
- The overall procedure is far too long to be reported here, but it is extensively explained in detail in the help online under the section *Device Characterization Wizard*
- What is important to be mentioned here is the built-in tool called *Sheetscan* (see next slide), very useful for capturing curves from vendor datasheets, which are needed during the device characterization process



Semiconductor Device Characterization

- **SheetScan**

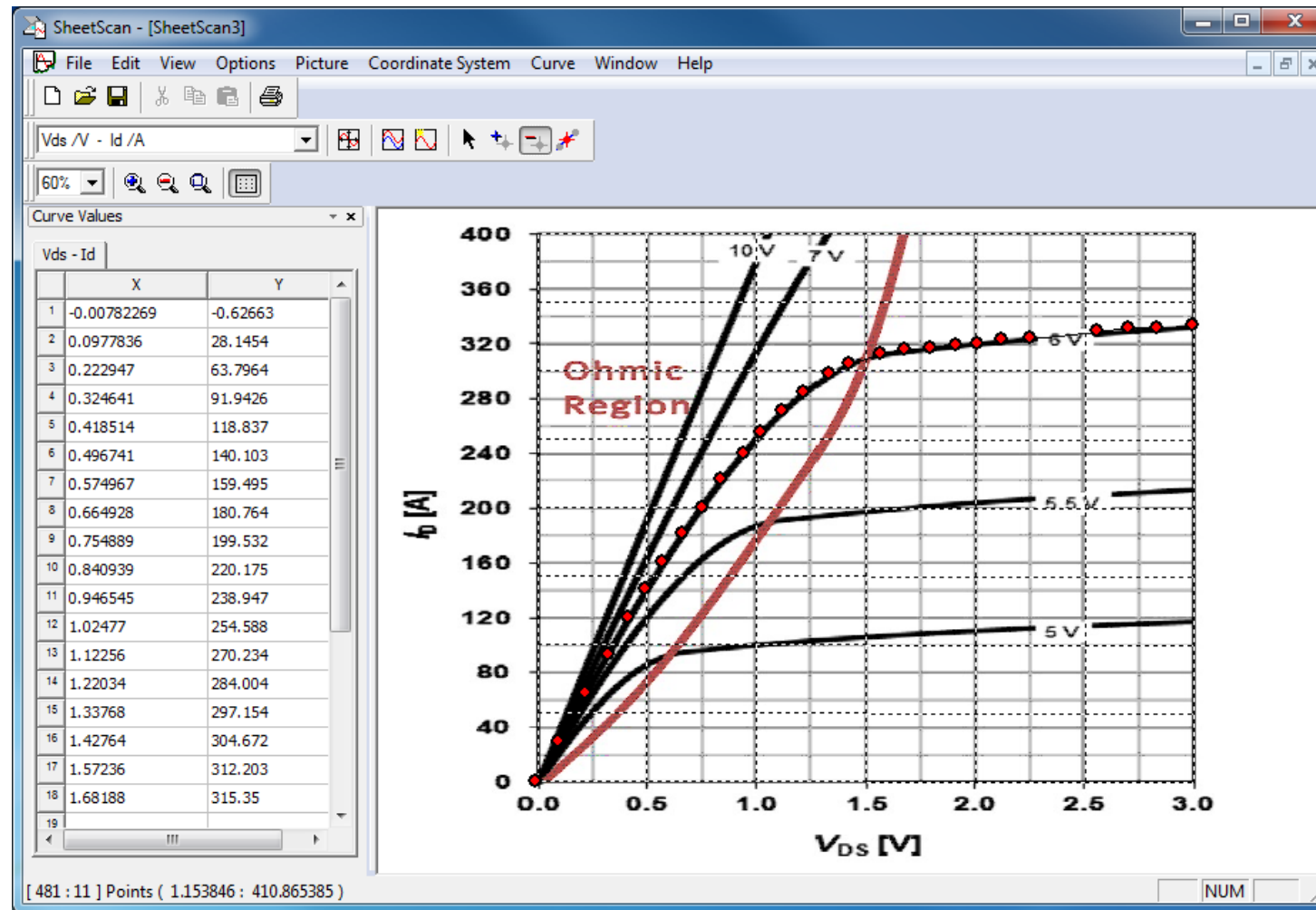
- SheetScan can be used to import a datasheet snapshot and then map axes on the image as an overlay. Users can then manually add datapoints to approximate the characteristic curve(s) of the datasheet
- Sampled data can then be converted to Simplorer format, and the extracted data exported to a Simplorer dataset or saved to a file
- SheetScan can be accessed from menu **Project** → **Datasets** and then clicking on the **SheetScan** button



- The process for creating a dataset using SheetScan includes four basic operations:
 - Loading a characteristics picture (*.jpeg format) into SheetScan
 - Defining a coordinate system for the imported datasheet picture
 - Defining a characteristic curve using the datasheet picture as reference
 - Exporting the characteristic curve data to a file or to a dataset

Semiconductor Device Characterization

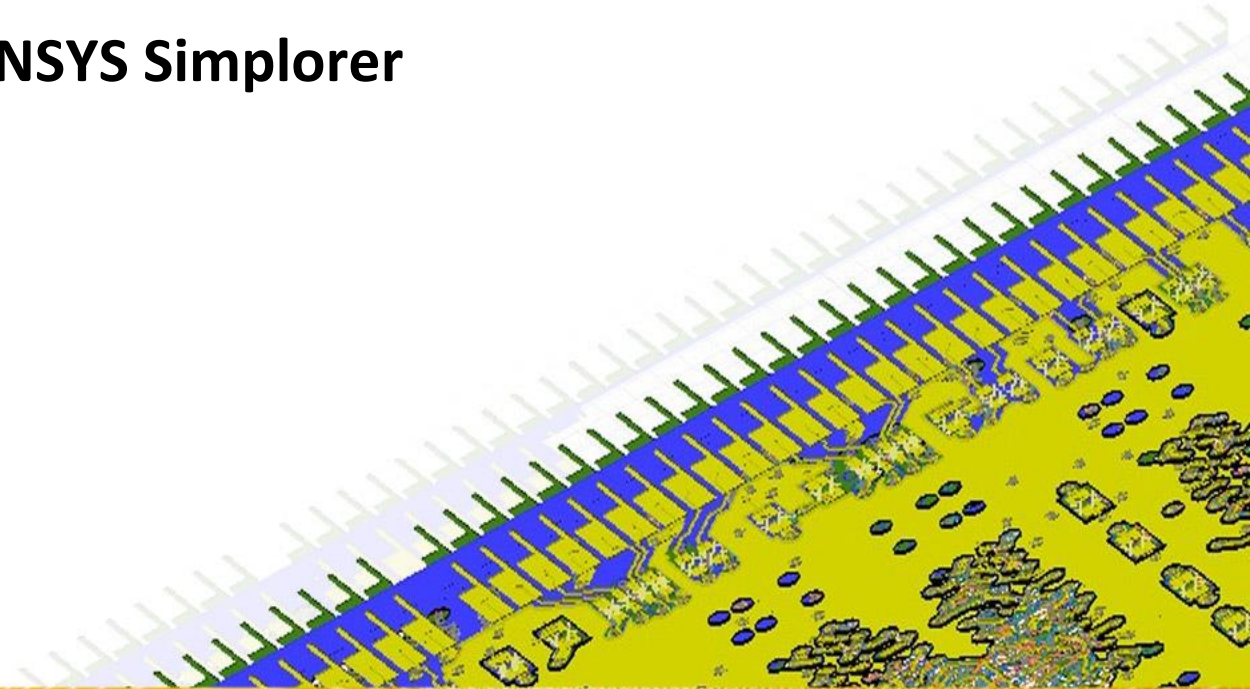
- Sheetscan Example





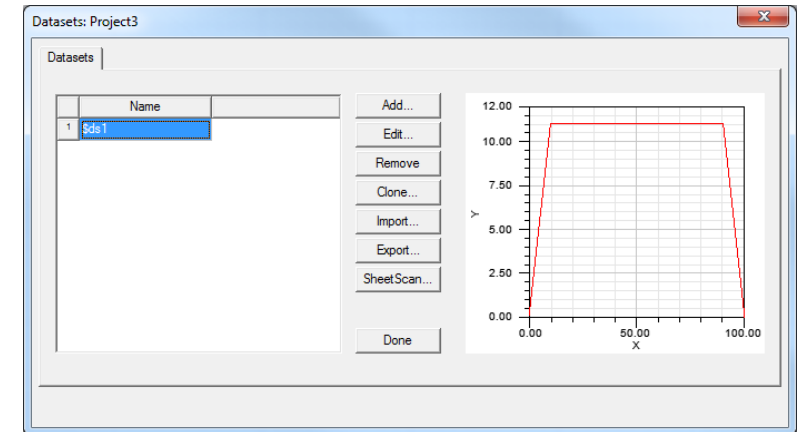
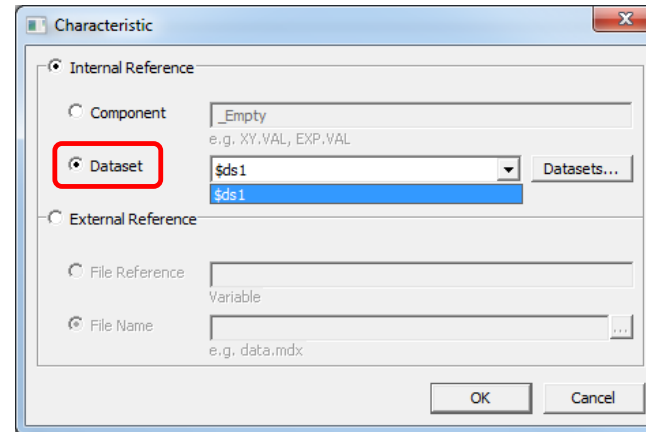
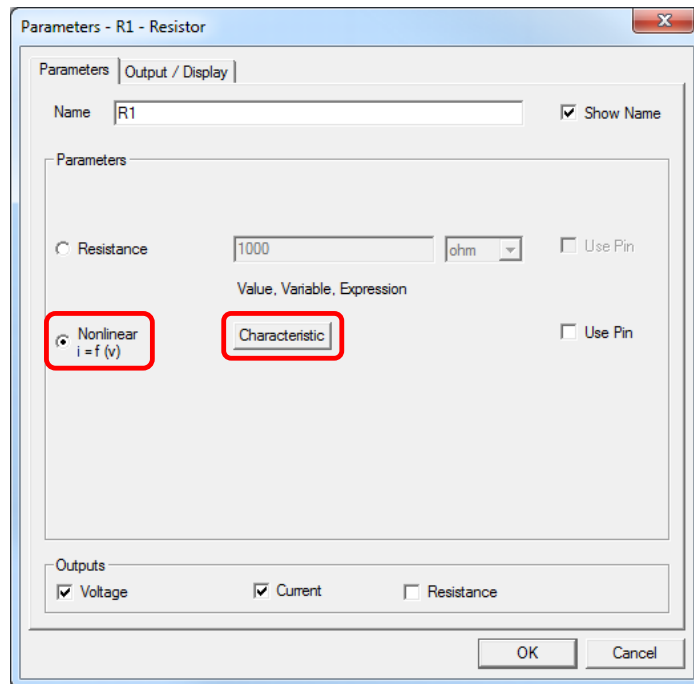
Non-Linear Components

Introduction to ANSYS Simplorer



Non-Linear Components

- Simplorer offers the possibility to characterize the behavior of nonlinear components in different ways
- One of the most used is through suitable **datasets**, which are defined with the aim to represent the non linear Output/Input dependency law
- Those **datasets** are then accessible by several components (DATAPAIRS, Resistors, Inductors, Voltmeters and many others) and define components behavior
- Here below an example of dataset used to define **Resistor current** vs **Resistor voltage** behavior



Summary

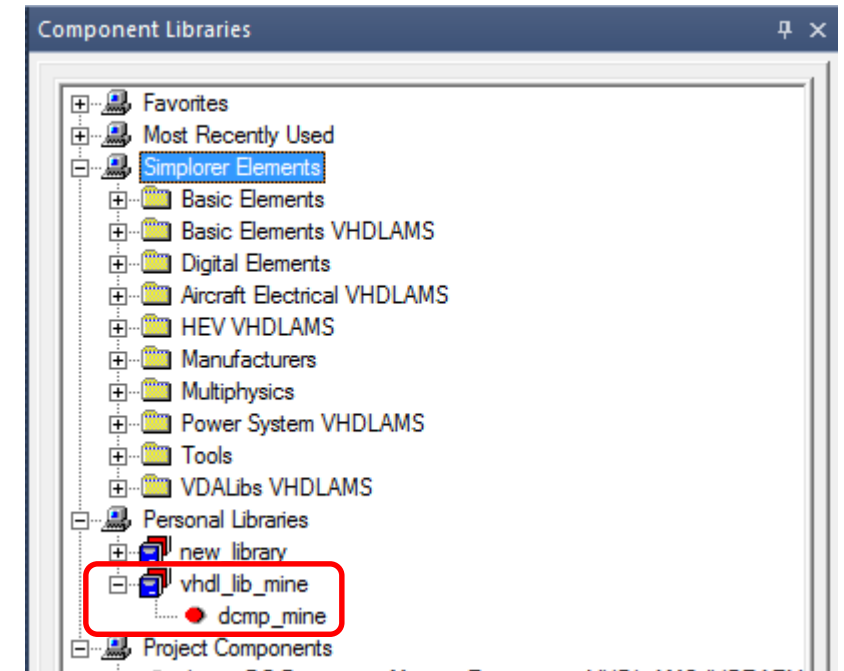
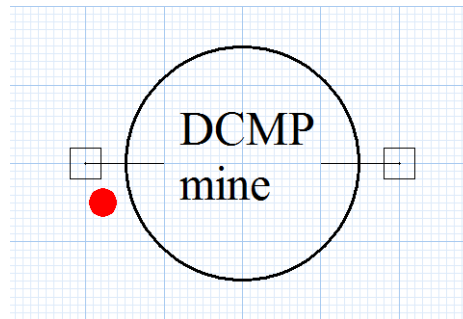
What have we learned in this session?

- **VHDL-AMS**
 - Language
 - Entity and Architecture(s)
- **Library Management**
- **Device Characterization**
 - Sheetscan
 - Characterization Workflow
- **Non-Linear Components**

Workshop 4.1 – VHDL Components and Export to Library

```
25
26 LIBRARY IEEE;
27 USE IEEE.ELECTRICAL_SYSTEMS.ALL;
28 USE IEEE.MECHANICAL_SYSTEMS.ALL;
29 USE IEEE.MATH_REAL.ALL;
30
31 ENTITY DCMp_mine IS
32   GENERIC(
33     la  : INDUCTANCE := 0.01;
34     ke  : REAL := 1.0;
35     j    : MOMENT_INERTIA := 0.075;
36     ia0  : CURRENT := 0.0;
37     n0   : REAL := 0.0;
38     phi0 : ANGLE := 0.0);
39   PORT(
40     TERMINAL n1, n2 : ELECTRICAL;
41     QUANTITY iout  : OUT CURRENT;
42     QUANTITY load  : IN TORQUE := 0.0;
43     QUANTITY ra    : IN RESISTANCE := 1.0;
44     QUANTITY n     : OUT REAL;
45     QUANTITY phi   : OUT ANGLE);
46 END ENTITY DCMp_mine;
```

dcmp_mine* arch: behav



Workshop 4.2 – Non-Linear Components

