SPICE Components in VHDL-AMS

Proposal for Discussion

Version: 0.6
Date: March 7, 2005
Summary

Based on the SPICE3 Version 3f3 User’s Manual and the Verilog-AMS Language Reference Manual 2.1 a proposal is made how to handle a selected number of basic SPICE elements in VHDL-AMS netlist descriptions. First versions of the proposal were discussed by the working group “Simulation of heterogeneous systems using VHDL-AMS” (VDA/FAT-AK30).

Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>January 10, 2004</td>
<td>First draft (in German)</td>
</tr>
<tr>
<td>0.2</td>
<td>January 15, 2004</td>
<td>Revision after discussion in the WG VDA/FAT-AK30</td>
</tr>
<tr>
<td>0.3</td>
<td>March 17, 2004</td>
<td>English translation</td>
</tr>
<tr>
<td>0.4</td>
<td>July 16, 2004</td>
<td>AC magnitude by default set to 0.0 in the declarations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of voltage and current source components,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPICE_LIB replaced by SPICE2VHD</td>
</tr>
<tr>
<td>0.5</td>
<td>September 1, 2004</td>
<td>Experiences from VHDL-AMS Library SPICE2VHD considered</td>
</tr>
<tr>
<td>0.6</td>
<td>March 7, 2005</td>
<td>Revised version after comments of E. Christen</td>
</tr>
</tbody>
</table>

Contact

Joachim Haase

E-Mail: joachim.haase@eas.iis.fhg.de

Address: Fraunhofer-Institut für Integrierte Schaltungen
Branch Lab Design Automation EAS
Zeunerstr. 38
D-01069 Dresden
1 Objectives

SPICE models are widely used to describe electrical networks. Therefore, it is desirable to re-use existing parametrized models, especially transistor level models, together with VHDL-AMS models. At the moment there does not exist a standard that defines how to include SPICE models in VHDL-AMS netlists. Essential ideas how to develop VHDL-AMS libraries were developed by E. Christen and K. Bakalar [1] and are considered in the following.

Interfaces of SPICE-like models that can be used in VHDL-AMS are proposed in the following. The proposal sets the following objectives:

- It should be possible to exchange a VHDL-AMS netlist with SPICE-like models between different simulation programs.
- The interfaces of SPICE-like models shall be specified.
- A base for the discussion of the implementation of SPICE-like models in commercial VHDL-AMS simulation tools shall be given.

The handling of SPICE subcircuits in VHDL-AMS is not addressed.

The interfaces are only specified for basic elements (see section 3). As a result of the discussion of this proposal the scope of models could be extended or reduced. SPICE-like current controlled voltage and current sources are knowingly not considered because they require an access to internal data objects of other models. A similar situation occurs in the case of coupled coils. The choice of models was done with respect to the models that shall be made available in Verilog-AMS (see [3], Appendix E SPICE compatibility, Table E.1).

There are similar problems concerning the usage of SPICE-like models in Verilog-AMS as in VHDL-AMS. Some of the difficulties that result from the missing standardization of SPICE netlist syntax are discussed in the current version of the Verilog-AMS Language Reference Manual (see [3], Appendix E). The presented proposal tries to take these problems into account. The following compromise is suggested:

- Parameters that directly characterise SPICE elements are named and used in accordance with SPICE3f3 (SPICE3f5 resp.).
- It should be possible to include existing technology-dependent parameters that are described by .MODEL cards in SPICE using a file interface. Thus, existing include files with .MODEL card information can be used in commercial tools.

2 Approach

Two packages

- SPICE_PARAMETERS and
- SPICE_COMPONENTS

are declared. They shall be compiled into a library symbolically named SPICE2VHD (see Appendix).
2.1 Package SPICE_PARAMETERS

The Package Header of SPICE_PARAMETERS declares:

- Types for the description of technology-dependent model parameters
  - RESISTOR_DATA
  - DIODE_DATA
  - BJT_DATA
  - JFET_DATA
  - MOSFET_DATA
  - MESFET_DATA

Data objects of these types (for instance constants) can carry the information of corresponding .MODEL cards in SPICE descriptions. A special implementation of these types is not required. To use a record type would be the best way. However with respect to implementation limits of some current simulators arrays are used in appendix A.1. It depends on the implementation whether array or records are used. The names are only place holders.

- Functions to initialize data objects depending on technology-dependent parameters
  - SET_RESISTOR_DATA
  - SET_DIODE_DATA
  - SET_BJT_DATA
  - SET_JFET_DATA
  - SET_MOSFET_DATA
  - SET_MESFET_DATA

Different versions of these functions shall be made available in the package body. It can be taken advantage of the possibility to overload functions.

  - Version 1 with the same model parameter notations as in SPICE3f3 [2]
    
    Because the parameters are initialised in the list of parameters in the function declaration it is not necessary to assign values to default parameters in a function call. If SPICE-like models are completely implemented in VHDL-AMS only this version is required.

  - Version 2 with passing of model parameters in a file
    
    This version supports the passing of the .MODEL card information using a file. This version makes it possible to pass parameters that are used in commercial simulation engines and differ from SPICE3f3. Several .MODEL cards can be stored in one file. One parameter of the function specifies the name of the card that has to be evaluated.

An example in section 4 illustrates the usage of these data types and functions.
2.2 Package SPICE_COMPONENTS

Interfaces of SPICE models are declared in the Package SPICE_COMPONENTS. They are based on the following decisions:

- Component names are used in accordance with the Verilog-AMS Language Reference Manual [3].
- Names of generic parameters are in accordance with SPICE3f3 [2].

The implementations of the models have to be compiled into a library symbolically named SPICE2VHD. The following requirements must be fulfilled:

- entity name and name of the associated component must be the same.
- The names of generic parameters and their default values must be equal in entity and component declaration.

Only one architecture must be developed for each entity. In the elaboration phase the last compiled architecture of a component is used. This allows to instantiate components without a configuration (null configuration). A direct instantiation that characterizes a design entity only by the entity name is also possible.

An example is given in section 4.

2.3 General conventions

During the initialisation of SPICE models partly simulation parameters like TSTEP and TSTOP from the .TRAN statement are used. These data are not available in VHDL-AMS models. How to handle these and similar cases the proposed consequences are summarized in the following table.

<table>
<thead>
<tr>
<th>Usage in SPICE</th>
<th>Implementation in VHDL-AMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A value must be assigned to the parameter.</td>
<td>Parameter is not initialized.</td>
</tr>
<tr>
<td>Parameter is not initialized. An initialisation is not required.</td>
<td>Real parameter are initialized using REAL'LOW. Undefined parameters of other types are not used.</td>
</tr>
<tr>
<td>Parameter is initialized with TSTEP.</td>
<td>If the parameter specifies a rising or falling time its value is set to 0.0. Otherwise it is required to map a value to the parameter during instantiation (see for example VEXP and IEXP).</td>
</tr>
<tr>
<td>Parameter is initialized with TSTOP.</td>
<td>Instead of TSTOP the value REAL'HIGH is used.</td>
</tr>
<tr>
<td>Parameter is initialized with 1/TSTOP.</td>
<td>If a parameter in SPICE3f3 is initialized using 1/TSTOP it is required to map a value to the parameter during instantiation (see for example VSINE).</td>
</tr>
<tr>
<td>A real-valued parameter is initialized with infinite.</td>
<td>The parameter is initialized with REAL'HIGH.</td>
</tr>
</tbody>
</table>
Parameter is initialized with the ambient temperature. The parameter is initialized with AMBIENT_TEMPERATURE of the Package MATERIAL_CONSTANTS of the IEEE library [4]. It is assumed that the value of AMBIENT_TEMPERATURE is given in degree Kelvin. In the VHDL-AMS models all temperatures will be given in Kelvin (instead of Celsius in SPICE).

Modell uses information of a .MODEL card. Assignment of a constant to the generic parameter MDATA of a model.

Possible initializations during quiescent domain analysis (using IC of OFF) are only considered in the models of capacitance and inductance.

## 3 Models

The following SPICE models must be made available in VHDL-AMS (see package SPICE_COMPONENTS). The names of the components are in accordance with the notations used in the Verilog-AMS proposal (see [3], Appendix E, Table E.1). The names of the parameters are to the greatest possible extent in accordance with the SPICE3 Version 3f3 User’s Manual [2].

### Basic elements

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESISTOR</td>
<td>Resistance [2], §3.1.1 and §3.1.2</td>
</tr>
<tr>
<td>CAPACITOR</td>
<td>Capacitance [2], §3.1.4</td>
</tr>
<tr>
<td>INDUCTOR</td>
<td>Inductance [2], §3.1.7</td>
</tr>
</tbody>
</table>

### Complex elements

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCVS</td>
<td>Voltage controlled voltage source [2], §3.2.2.2</td>
</tr>
<tr>
<td>VCCS</td>
<td>Voltage controlled current source [2], §3.2.2.1</td>
</tr>
<tr>
<td>TLINE</td>
<td>Lossless line [2], §3.3.1</td>
</tr>
</tbody>
</table>

### Independent voltage sources

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDC</td>
<td>Constant voltage source [2], §3.2.1</td>
</tr>
<tr>
<td>VEXP</td>
<td>EXP voltage source [2], §3.2.1/§3.2.1.3</td>
</tr>
<tr>
<td>VPULSE</td>
<td>PULSE voltage source [2], §3.2.1/§3.2.1.1</td>
</tr>
<tr>
<td>VPWL</td>
<td>PWL voltage source [2], §3.2.1/§3.2.1.4</td>
</tr>
<tr>
<td>VSINE</td>
<td>Sinewave voltage source [2], §3.2.1/§3.2.1.2</td>
</tr>
<tr>
<td>VSFFM</td>
<td>FM modulated voltage source [2], §3.2.1/§3.2.1.5</td>
</tr>
</tbody>
</table>

### Independent current sources

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDC</td>
<td>Constant current source [2], §3.2.1</td>
</tr>
<tr>
<td>IEXP</td>
<td>EXP current source [2], §3.2.1/§3.2.1.3</td>
</tr>
<tr>
<td>IPULSE</td>
<td>PULSE current source [2], §3.2.1/§3.2.1.1</td>
</tr>
<tr>
<td>IPWL</td>
<td>PWL current source [2], §3.2.1/§3.2.1.4</td>
</tr>
<tr>
<td>ISINE</td>
<td>Sinewave current source [2], §3.2.1/§3.2.1.2</td>
</tr>
<tr>
<td>ISFFM</td>
<td>FM modulated current source [2], §3.2.1/§3.2.1.5</td>
</tr>
</tbody>
</table>
Semiconductor models

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIODE</td>
<td>Diode [2], §3.4.1</td>
</tr>
<tr>
<td>BJT</td>
<td>Bipolar transistor [2], §3.4.2</td>
</tr>
<tr>
<td>JFET</td>
<td>J-field effect transistor [2], §3.4.5</td>
</tr>
<tr>
<td>MOSFET</td>
<td>MOSFET [2], §3.4.7</td>
</tr>
<tr>
<td>MESFET</td>
<td>MESFET [2], §3.4.9</td>
</tr>
</tbody>
</table>

4 Example

Figure 1 shows the circuit that is used in the following and will be described in different ways. The same circuit was also used to demonstrate the handling of SPICE elements in Verilog-AMS (see [3]). The circuit is described by the following SPICE netlist:

```
.MODEL VERTNPN NPN BF=80 IS=1E-18 RB=100 VAF=50
+ CJB=3PF CJC=2PF CJS=2PF TF=0.3NS TR=6NS

.SUBCKT diffPair c1 b1 e c2 b2
Q1 c1 b1 e vertNPN
Q2 c2 b2 e vertNPN
.ENDS
```

VHDL-AMS description (Version 1)

The technology-dependent data of the .MODEL card are assigned to a constant vertNPN using the function SET_BJT_DATA of the package SPICE_PARAMETERS. The parameter notification is in accordance with SPICE3f3. The architecture a0 describes the structure of the circuit using VHDL-AMS. The component BJT is declared in the package SPICE_COMPONENTS. By default the substrat terminal of the bipolar transistor is connected to the electrical reference in SPICE. An explicit assignment of a node to this terminal is not required in SPICE. Such a default assignment is not possible in VHDL-AMS. The substrat terminal has to be explicitly connected to ELECTRICAL_REF. Following the presented proposal the VHDL-AMS description of the circuit looks like:

```
library IEEE, SPICE2VHD;
use IEEE.ELECTRICAL_SYSTEMS.all;
use SPICE2VHD.SPICE_PARAMETERS.all;
use SPICE2VHD.SPICE_COMPONENTS.all;

entity diffPair is
  port ( 
    terminal cl, bl, e, c2, b2 : ELECTRICAL );
end entity diffPair;
architecture a0 of diffPair is
```
constant vertNPN : BJT_DATA := SET_BJT_DATA (MODEL => NPN, BF => 80.0, ISS => 1.0E-18, RB => 100.0, VAF => 50.0, CJE => 3.0E-12, CJC => 2.0E-12, CJ5 => 2.0E-12, TF => 0.3E-9, TR => 6.0E-9);

begin

Q1: BJT generic map (vertNPN)
    port map (c1, b1, e, electrical_ref);

Q2: BJT generic map (MDATA => vertNPN)
    port map (NC => c2, NB => b2, NE => e, NS => electrical_ref);

end architecture a0;

VHDL-AMS description (Version 2)

It is assumed that the .MODEL card information is saved in a file modeldata.inc (for instance). This file contains the following lines:

.MODEL VERTNPN NPN BF=80 IS=1E-18 RB=100 VAF=50 + CJB=3PF CJC=2PF CJS=2PF TF=0.3NS TR=6NS

The file can be included in a SPICE netlist (see [3], §2.5):

.include modeldata.inc
.SUBCKT diffPair c1 b1 e c2 b2
Q1 c1 b1 e vertNPN
Q2 c2 b2 e vertNPN
.ENDS

It would be helpful to use the same file to specify technology-dependent data in VHDL-AMS. This could be done with an overloaded function SET_BJT_DATA as suggested in the package SPICE_PARAMETERS. The architecture a1 demonstrates the usage of the overloaded function.

architecture a1 of diffPair is

-- Make the values of the model card in a file available

constant vertNPN : BJT_DATA := SET_BJT_DATA (MNAME => "VERTNPN", FILENAME => "modeldata.inc" );

begin

Q1: BJT generic map (vertNPN)
    port map (c1, b1, e, electrical_ref);

Q2: BJT generic map (vertNPN)
    port map (c2, b2, e, electrical_ref);

end architecture a1;
5 Proposal for further steps and problems

It seems that the following procedure could help to get more clarity about the SPICE components in VHDL-AMS

- Discussion of the possibilities to realize the suggested functionality in their tools with EDA vendors
- Discussion of the proposal with other potential interested parties (BEAMS, IEEE WG 1076.1)

From the current point of view there is not a satisfying solution for the following problems

- Handling of MOSFET models LEVEL 4 and LEVEL 5 (BSIM)
  This point influences the specification of the function SET_MOSFET_DATA (Package SPICE_PARAMETERS).
- Consideration of the activities to define so-called „built-in“ models of semiconductor devices using Verilog-A/Verilog-AMS [5]
- Numbering (design) of error messages

References


Appendix

A1. Package SPICE_PARAMETERS

library IEEE;
use IEEE.ELECTRICAL_SYSTEMS.all;
use IEEE.MATERIAL_CONSTANTS.all;

package SPICE_PARAMETERS is

  -- characterization of simulation program

  constant SPICE_SIMULATOR : STRING := "SPICE3F";

  -- nominal temperature

  Constant SPICE_TNOM      : REAL   := 300.15;   -- nominal temperature (in Kelvin)

  -- Spice device models (SUM §2.3)

  type MODEL_TYPE is
    UNDEF,        -- model type is not defined
    R,            -- semiconductor resistor model
    C,            -- semiconductor capacitor model
    SW,           -- voltage controlled switch
    CSW,          -- current controlled switch
    URC,          -- uniform distributed rc model
    LTRA,         -- lossy transmission line model
    D,            -- diode model
    NPN,          -- npn BJT model
    PNP,          -- pnp BJT model
    NJF,          -- n-channel JFET model
    PJF,          -- p-channel JFET model
    NMOS,         -- n-channel MOSFET model
    PMOS,         -- p-channel MOSFET model
    NMF,          -- n-channel MESFET model
    PMF           -- p-channel MESFET model
  );

  -- Types, constants, and functions for resistor model (SUM §3.1.3)
type RESISTOR_DATA is array (1 to 6) of REAL; -- implementation dependent

function SET_RESISTOR_DATA (constant RSH    : REAL ;             -- sheet resistance (ohm/sqr)
countant TC1    : REAL := 0.0;       -- first order temperature coefficient
constant TC2    : REAL := 0.0;       -- second order temperature coefficient
constant DEFW   : REAL := 1.0E-6;    -- default width (in meters)
countant NARROW : REAL := 0.0;       -- narrowing due to side etching
constant TNOM   : REAL := SPICE_TNOM    -- parameter measurement temperature
) return RESISTOR_DATA;

function SET_RESISTOR_DATA (constant MNAME     : STRING;         -- model card searched in FILENAME
countant FILENAME  : STRING ) return RESISTOR_DATA;

constant DEFAULT_RESISTOR_DATA : RESISTOR_DATA;
-- assignment in package body

-- Types, constants, and functions for capacitor model (SUM §3.1.4)

type CAPACITOR_DATA is array (1 to 4) of REAL;     -- implementation dependent

function SET_CAPACITOR_DATA (constant CJ     : REAL := REAL'LOW;     -- junction bottom capacitance (in
constant CJSW   : REAL := REAL'LOW;     -- junction sidewall capacitance (in
constant DEFW   : REAL := 1.0E-6;    -- default width (in m)
countant NARROW : REAL := 0.0           -- narrowing due to side etching (in m)
) return CAPACITOR_DATA;

constant DEFAULT_CAPACITOR_DATA : CAPACITOR_DATA;
-- assignment in package body

-- Types, constants, and functions for diode model (SUM §3.4.2)

type DIODE_DATA is array (1 to 15) of REAL;     -- implementation dependent

function SET_DIODE_DATA (constant ISS    : REAL := 1.0E-14;   -- saturation current (A)
countant RS     : REAL := 0.0;       -- ohmic resistance (ohm)
countant N      : REAL := 1.0;       -- emission coefficient
constant TT     : REAL := 0.0;       -- transit time (in seconds)
countant CJ0    : REAL := 0.0;       -- zero-bias junction capacitance (in F)
countant VJ     : REAL := 1.0;       -- junction potential (in V)
countant M      : REAL := 0.5;       -- grading coefficient
constant EG     : REAL := 1.11;      -- activation energy (in eV)
countant XTI    : REAL := 3.0;       -- saturation-current temp. exp
constant KF     : REAL := 0.0;       -- flicker noise coefficient
constant AF     : REAL := 1.0;       -- flicker noise exponent
constant FC     : REAL := 0.5;       -- coefficient for forward-bias
constant BV     : REAL := REAL'HIGH; -- reverse breakdown voltage (in V)
countant IBV    : REAL := 1.0E-3;    -- current at breakdown voltage (in A)
countant TNOM   : REAL := SPICE_TNOM    -- parameter measurement temperature
) return DIODE_DATA;

function SET_DIODE_DATA (constant MNAME     : STRING;         -- model card searched in FILENAME
countant FILENAME  : STRING ) return DIODE_DATA;

-- Types, constants, and functions for BJT model (SUM §3.4.4)
type BJT_DATA is array (1 to 42) of REAL; -- implementation dependent

function SET_BJT_DATA (
  constant MODEL : MODEL_TYPE; -- type of BJT (NPN|PNP)
  constant ISS : REAL := 1.0E-14; -- saturation current (in A)
  constant BF : REAL := 100.0; -- ideal maximum forward BETA
  constant NF : REAL := 1.0; -- forward current emission coefficient
  constant VAF : REAL := REAL’HIGH; -- forward Early voltage (in V)
  constant IKF : REAL := REAL’HIGH; -- corner for forward BETA high current
  constant ISE : REAL := 0.0; -- B-E leakage saturation current (in A)
  constant NE : REAL := 1.5; -- B-E leakage emission coefficient
  constant BR : REAL := 1.0; -- ideal maximum reverse BETA
  constant NR : REAL := 1.0; -- reverse current emission coefficient
  constant VAR : REAL := REAL’HIGH; -- reverse Early voltage (in V)
  constant IKR : REAL := REAL’HIGH; -- corner for reverse BETA high current
  constant ISC : REAL := 0.0; -- B-C leakage saturation current (in A)
  constant NC : REAL := 2.0; -- B-C leakage emission coefficient
  constant RB : REAL := 0.0; -- zero bias base resistance (in ohm)
  constant IRB : REAL := REAL’HIGH; -- current where base resistance
  constant RBM : REAL := REAL’HIGH; -- minimum base resistance at high
  constant RE : REAL := 0.0; -- emitter resistance (in ohm)
  constant RC : REAL := 0.0; -- collector resistance (in ohm)
  constant CJE : REAL := 0.0; -- B-E zero-bias depletion capacitance
  constant VJE : REAL := 0.75; -- B-E built-in potential (in V)
  constant MJE : REAL := 0.33; -- B-E junction exponential factor
  constant TF : REAL := 0.0; -- ideal forward transit time
  constant XTF : REAL := 0.0; -- coefficient for bias dependence of TF
  constant VTF : REAL := REAL’HIGH; -- voltage describing VBC dependence
  constant ITF : REAL := 0.0; -- high-current parameter for effect
  constant PTF : REAL := 0.0; -- excess phase at freq=1.0/(TF*2PI) Hz
  constant CJC : REAL := 0.0; -- B-C zero-bias depletion capacitance
  constant JVC : REAL := 0.75; -- B-C built-in potential (in V)
  constant MJC : REAL := 0.33; -- B-C junction exponential factor
  constant XCJC : REAL := 1.0; -- fraction B-C depletion capacitance
  constant TR : REAL := 0.0; -- ideal reverse transit time
  constant CJS : REAL := 0.0; -- zero-bias collector-substrate
  constant VJS : REAL := 0.75; -- substrate built-in potential (in V)
  constant MJS : REAL := 0.0; -- substrate junction exponential factor
  constant XTB : REAL := 0.0; -- forward and reverse beta
  constant EG : REAL := 1.11; -- energy gap for temperature
  constant XTI : REAL := 3.0; -- temperature exponent for effect
  constant KF : REAL := 0.0; -- flicker noise coefficient
  constant AF : REAL := 1.0; -- flicker noise exponent
  constant FC : REAL := 0.5; -- coefficient for forward-bias
  constant CJS : REAL := 0.0; -- zero-bias collector-substrate
  constant XTI : REAL := 3.0; -- temperature exponent for effect
  constant TNI : REAL := REAL’HIGH; -- parameter measurement temperature
  constant TNOM : REAL := SPICE_TNOM -- parameter measurement temperature
) return BJT_DATA;

-- Types, constants, and functions for JFET model (SUM §3.4.6)
type JFET_DATA is array (1 to 15) of REAL; -- implementation dependent

function SET_JFET_DATA (  
constant MODEL : MODEL_TYPE; -- type of JFET (njf|pjf)  
constant VT0 : REAL := -2.0; -- threshold voltage (in V)  
constant BETAD : REAL := 1.0E-4; -- transconductance parameter  
constant LAMBDA : REAL := 0.0; -- channel-length modulation parameter  
constant RD : REAL := 0.0; -- drain ohmic resistance (in ohm)  
constant RS : REAL := 0.0; -- source ohmic resistance (in ohm)  
constant CGS : REAL := 0.0; -- zero-bias G-S junction capacitance  
constant CGD : REAL := 0.0; -- zero-bias G-D junction capacitance  
constant PB : REAL := 1.0; -- gate junction potential (in V)  
constant ISS : REAL := 1.0E-14; -- gate saturation current (in A)  
constant B : REAL := 1.0; -- doping tail parameter  
constant KF : REAL := 0.0; -- flicker noise coefficient  
constant AF : REAL := 1.0; -- flicker noise exponent  
constant FC : REAL := 0.5; -- coefficient for forward-bias depletion capacitance formula  
constant TNOM : REAL := SPICE_TNOM -- parameter measurement temperature -- (in Kelvin)  
) return JFET_DATA;

function SET_JFET_DATA (  
constant MNAME : STRING; -- model card searched in FILENAME  
constant FILENAME : STRING  
) return JFET_DATA;

-- Types, constants, and functions for MOSFET model (SUM §3.4.8)

type MOSFET_DATA is array (1 to 52) of REAL; -- implementation dependent

function SET_MOSFET_DATA (  
constant MODEL : MODEL_TYPE; -- type of MOSFET (NMOS|PMOS)  
constant LEVEL : positive := 1; -- model index  
constant VT0 : REAL := 0.0; -- threshold voltage (in V)  
constant KF : REAL := 2.0E-5; -- transconductance parameter  
constant GAMMA : REAL := 0.0; -- bulk threshold parameter  
constant PHI : REAL := 0.6; -- surface potential (in V)  
constant LAMBDA : REAL := 0.0; -- channel-length modulation (only MOS1 and MOS2)  
constant RD : REAL := 0.0; -- drain ohmic resistance (in ohm)  
constant RS : REAL := 0.0; -- source ohmic resistance (in ohm)  
constant CBD : REAL := 0.0; -- zero-bias B-D junction capacitance  
constant CBS : REAL := 0.0; -- zero-bias B-S junction capacitance  
constant CGSO : REAL := 0.0; -- gate-source overlap capacitance per meter channel width (in F/m)  
constant CGDO : REAL := 0.0; -- gate-drain overlap capacitance per meter channel width (in F/m)  
constant CGBO : REAL := 0.0; -- gate-bulk overlap capacitance per meter channel width (in F/m)  
constant RSH : REAL := 0.0; -- drain and source diffusion sheet resistance (in ohm/sqr)  
constant CJ : REAL := 0.0; -- zero-bias bulk junction bottom cap. per sq-meter of junction area -- (in F/m**2)  
constant MJ : REAL := 0.5; -- bulk junction bottom grading coeff.  
constant CJSW : REAL := 0.0; -- zero-bias bulk sidewall cap. per sq-meter of junction area -- (in F/m**2)  
) return MOSFET_DATA;
constant MJSW   : REAL := REAL'LOW;    -- bulk junction sidewall grading
  -- coefficient
  -- 0.50 for level1
  -- 0.33 for level2, 3
constant JS     : REAL := REAL'LOW;    -- bulk junction saturation current
  -- per sq-meter of junction area
  -- (in A/m**2)
constant TOX    : REAL := 1.0E-7;      -- oxide thickness (in meter)
constant NSUB   : REAL := 0.0;         -- substrate doping (in 1/cm**2)
constant NSS    : REAL := 0.0;         -- surface state density (in 1/cm**2)
constant NFS    : REAL := 0.0;         -- fast surface state density
constant TPG    : REAL := 1.0;         -- type of gate material
  -- +1 opp. to substrate
  -- -1 same as substrate
  -- 0 Al gate
constant XJ     : REAL := 0.0;         -- metallurgical junction depth
  -- (in meter)
constant LD     : REAL := 0.0;         -- lateral diffusion (in meter)
constant U0     : REAL := 600.0;       -- surface mobility (in cm**2/Vs)
constant UCRIT  : REAL := 1.0E4;       -- critical field for mobility
  -- degration (MOS2 only) (in V/cm)
constant UEXP   : REAL := 0.0;         -- critical field exponent in
  -- mobility degradation (MOS2 only)
constant UTRA   : REAL := 0.0;         -- transverse field coeff. (mobility)
  -- (deleted for MOS2)
constant VMAX   : REAL := 0.0;         -- maximum drift velocity of carriers
  -- (in m/s)
constant NEFF   : REAL := 1.0;         -- total channel-charge
  -- (fixed and mobile)
  -- coefficient (MOS2 only)
constant KF     : REAL := 0.0;         -- flicker noise coefficient
constant AF     : REAL := 1.0;         -- flicker noise exponent
constant FC     : REAL := 0.5;         -- coefficient for forward-bias
  -- depletion capacitance formula
constant DELTA  : REAL := 0.0;         -- width effect on threshold voltage
  -- (MOS2 and MOS3)
constant THETA  : REAL := 0.0;         -- mobility modulation (MOS3 only)
  -- (in 1/V)
constant ETA    : REAL := 0.0;         -- static feedback (MOS3 only)
constant KAPPA  : REAL := 0.2;         -- saturation field factor (MOS3 only)
constant TNOM   : REAL := SPICE_TNOM   -- parameter measurement temperature
  -- (in Kelvin)
) return MOSFET_DATA;

function SET_MOSFET_DATA ( constant MNAME     : STRING;           -- model card searched in FILENAME
  constant FILENAME  : STRING ) return MOSFET_DATA;

-- Types, constants, and functions for MESFET model (SUM §3.4.10)
type MESFET_DATA is array (1 to 14) of REAL;     -- implementation dependent

function SET_MESFET_DATA ( constant MODEL  : MODEL_TYPE;          -- type of MESFET (NMF|PMF)
  constant VT0    : REAL := 0.0;         -- threshold voltage (in V)
  constant BETA   : REAL := 1.0E-4;      -- transconductance parameter
  -- (in A/V**2)
  constant B      : REAL := 0.3;         -- doping tail extending parameter
  -- (in 1/V)
  constant ALPHA  : REAL := 2.0;         -- saturation voltage parameter
  -- (in 1/V)
  constant LAMBDA : REAL := 0.0;         -- channel-length modulation
  constant RD     : REAL := 0.0;         -- drain ohmic resistance (in ohm)
  constant RS     : REAL := 0.0;         -- source ohmic resistance (in ohm)
  constant CGS    : REAL := 0.0;         -- zero-bias G-S junction capacitance
  -- (in F)
  constant CGD    : REAL := 0.0;         -- zero-bias G-D junction capacitance
  -- (in F)
  constant PB     : REAL := 0.8;         -- bulk junction potential (in V)
) return MESFET_DATA;
constant KF     : REAL := 0.0;         -- flicker noise coefficient
constant AF     : REAL := 1.0;         -- flicker noise exponent
constant FC     : REAL := 0.5          -- coefficient for forward-bias
                     -- depletion capacitance formula
) return MOSFET_DATA;

function SET_MESFET_DATA (
    constant MNAME     : STRING;           -- model card searched in FILENAME
    constant FILENAME  : STRING
) return MESFET_DATA;

end package SPICE_PARAMETERS;
A2. Package SPICE_COMPONENTS

-- BASIC SPICE ELEMENTS ------------------------------------------------------
-- resistor (SUM §3.1.1, §3.1.2)

component RESISTOR
  generic
    R : REAL := REAL'LOW;               -- resistance (in ohms)
    MDATA : RESISTOR_DATA := DEFAULT_RESISTOR_DATA; -- model data
    L : REAL := REAL'LOW;               -- length (in meters)
    W : REAL := REAL'LOW;               -- width (in meters)
    TEMP : REAL := AMBIENT_TEMPERATURE; -- initial condition
  end generic;
  port
    ( terminal P : ELECTRICAL;
      terminal N : ELECTRICAL );
end component RESISTOR;

-- capacitor (SUM §3.1.4)

component CAPACITOR
  generic
    C : REAL := REAL'LOW;               -- capacitance (in F)
    MDATA : CAPACITOR_DATA := DEFAULT_CAPACITOR_DATA; -- model data
    L : REAL := REAL'LOW;               -- length (in meters)
    W : REAL := REAL'LOW;               -- width (in meters)
    IC : REAL := REAL'LOW;              -- initial condition
  end generic;
  port
    ( terminal P : ELECTRICAL;
      terminal N : ELECTRICAL );
end component CAPACITOR;
-- inductor (SUM §3.1.4)

component INDUCTOR
  generic
    L : REAL;  -- inductance (in H)
    IC : REAL := REAL'LOW  -- (optional) initial condition
  );
  port
    (terminal P : ELECTRICAL;
     terminal N : ELECTRICAL
  );
end component INDUCTOR;

-- COMPLEX SPICE ELEMENTS ---------------------------------------------------

-- voltage-controlled voltage source (SUM §3.2.2.2)

component VCVS
  generic
    GAIN : REAL;  -- voltage gain
  );
  port
    (terminal P : ELECTRICAL;
     terminal N : ELECTRICAL;
     terminal PS : ELECTRICAL;
     terminal NS : ELECTRICAL
  );
end component VCVS;

-- voltage-controlled current source (SUM §3.2.2.1)

component VCCS
  generic
    GM : REAL;  -- transconductance (in mhos)
  );
  port
    (terminal P : ELECTRICAL;
     terminal N : ELECTRICAL;
     terminal PS : ELECTRICAL;
     terminal NS : ELECTRICAL
  );
end component VCCS;

-- lossless transmission line (SUM §3.3.1)

component TLINE
  generic
    Z0 : REAL;  -- characteristic impedance
    TD : REAL := REAL'LOW;  -- delay time (in seconds)
    F : REAL := REAL'LOW;  -- frequency (in Hz)
    NL : REAL := REAL'LOW  -- normalized ELECTRICAL length
  );
  port
    (terminal N1 : ELECTRICAL;
     terminal N2 : ELECTRICAL;
     terminal N3 : ELECTRICAL;
     terminal N4 : ELECTRICAL
  );
end component TLINE;
-- INDEPENDENT VOLTAGE SOURCES -----------------------------------------------

-- Constant voltage source (SUM §3.2.1)

component VDC
generic  (
  DC      : REAL ;                -- DC value
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component VDC;

-- exp voltage source (SUM §3.2.1/§3.2.1.3)

component VEXP
generic  (
  V1      : REAL;                 -- initial value (in V)
  V2      : REAL;                 -- pulsed value (in V)
  TD1     : REAL := 0.0;          -- rise delay time (in seconds)
  TAU1    : REAL;                 -- rise time constant (in seconds)
  TD2     : REAL;                 -- fall delay time (in seconds)
  TAU2    : REAL;                 -- fall time constant
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component VEXP;

-- pulse voltage source (SUM §3.2.1/§3.2.1.1)

component VPULSE
generic  (
  V1      : REAL;                 -- initial value (in V)
  V2      : REAL;                 -- pulsed value (in V)
  TD      : REAL := 0.0;          -- delay time (in seconds)
  TR      : REAL := 0.0;          -- rise time (in seconds)
  TF      : REAL := 0.0;          -- fall time (in seconds)
  PW      : REAL := REAL'HIGH;    -- pulse width (in seconds)
  PER     : REAL := REAL'HIGH;    -- period (in seconds)
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component VPULSE;

-- pwI voltage source (SUM §3.2.1/§3.2.1.4)

component VPWL
generic  (
  WAVE    : REAL_VECTOR;          -- time value pairs T1, V1 <,T2, V2, ...>
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component VPWL;
-- sinusoidal voltage source (SUM §3.2.1/§3.2.1.2)

component VSINE
generic  (
  VO      : REAL;                 -- offset (in V)
  VA      : REAL;                 -- amplitude (in V)
  FREQ    : REAL;                 -- frequency (in Hz)
  TD      : REAL := 0.0;          -- delay (in seconds)
  THETA   : REAL := 0.0;          -- damping factor (in 1/seconds)
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component VSINE;

-- single frequency fm voltage source (SUM §3.2.1/§3.2.1.2)

component VSFFM
generic  (
  VO      : REAL;                 -- offset (in V)
  VA      : REAL;                 -- amplitude (in V)
  FC      : REAL;                 -- carrier frequency (in Hz)
  MDI     : REAL;                 -- modulation index
  FS      : REAL;                 -- signal frequency (in Hz)
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component VSFFM;

-- INDEPENDENT CURRENT SOURCES ----------------------------------------------

-- Constant current source (SUM §3.2.1)

component IDC
generic  (
  DC      : REAL ;                -- DC value
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component IDC;

-- exp current source (SUM §3.2.1/§3.2.1.3)

component IEXP
generic  (
  V1      : REAL;                 -- initial value (in A)
  V2      : REAL;                 -- pulsed value (in A)
  TD1     : REAL := 0.0;          -- rise delay time (in seconds)
  TAU1    : REAL;                 -- rise time constant (in seconds)
  TD2     : REAL;                 -- fall delay time (in seconds)
  TAU2    : REAL;                 -- fall time constant (in seconds)
  ACMAG   : REAL := 0.0;          -- AC magnitude
  ACPHASE : REAL := 0.0           -- AC phase
);
port     (
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component IEXP;

-- pulse current source (SUM §3.2.1/§3.1.1.1)
component IPULSE
  generic (  
    V1      : REAL;                 -- initial value (in A)  
    V2      : REAL;                 -- pulsed value (in A)  
    TD      : REAL := 0.0;          -- delay time (in seconds)  
    TR      : REAL := 0.0;          -- rise time (in seconds)  
    TF      : REAL := 0.0;          -- fall time (in seconds)  
    PW      : REAL := REAL'HIGH;    -- pulse width (in seconds)  
    PER     : REAL := REAL'HIGH;    -- period (in seconds)  
    ACMAG   : REAL := 0.0;          -- AC magnitude  
    ACPHASE : REAL := 0.0           -- AC phase  
  );
  port     (  
    terminal P : ELECTRICAL;  
    terminal N : ELECTRICAL  
  );
end component IPULSE;

-- pwl current source (SUM §3.2.1/§3.2.1.4)
component IPWL
  generic (  
    WAVE    : REAL_VECTOR;          -- time value pairs T1, V1 <,T2, V2, ...>  
    ACMAG   : REAL := 0.0;          -- AC magnitude  
    ACPHASE : REAL := 0.0           -- AC phase  
  );
  port     (  
    terminal P : ELECTRICAL;  
    terminal N : ELECTRICAL  
  );
end component IPWL;

-- sinusoidal current source (SUM §3.2.1/§3.2.1.2)
component ISINE
  generic (  
    VO      : REAL;                 -- offset (in A)  
    VA      : REAL;                 -- amplitude (in A)  
    FREQ    : REAL;                 -- frequency (in Hz)  
    TD      : REAL := 0.0;          -- delay (in seconds)  
    THETA   : REAL := 0.0;          -- damping factor (in 1/seconds)  
    ACMAG   : REAL := 0.0;          -- AC magnitude  
    ACPHASE : REAL := 0.0           -- AC phase  
  );
  port     (  
    terminal P : ELECTRICAL;  
    terminal N : ELECTRICAL  
  );
end component ISINE;
-- single frequency fm current source (SUM §3.2.1.2)

component ISFFM
generic (
  VO : REAL; -- offset (in A)
  VA : REAL; -- amplitude (in A)
  FC : REAL; -- carrier frequency (in Hz)
  MDI : REAL; -- modulation index
  FS : REAL; -- signal frequency (in Hz)
  ACMAG : REAL := 0.0; -- AC magnitude
  ACPHASE : REAL := 0.0 -- AC phase
);
port ( 
  terminal P : ELECTRICAL;
  terminal N : ELECTRICAL
);
end component ISFFM;

-- diode (SUM §3.4.1)

component DIODE
generic ( 
  MDATA : DIODE_DATA ; -- model data
  AREA  : REAL       := 1.0;  -- area factor
  TEMP  : REAL       := AMBIENT_TEMPERATURE
);
port ( 
  terminal A : ELECTRICAL;
  terminal C : ELECTRICAL
);
end component DIODE;

-- BJT (SUM §3.4.3)

component BJT
generic ( 
  MDATA  : BJT_DATA ; -- model data
  AREA   : REAL       := 1.0;  -- area factor
  TEMP   : REAL       := AMBIENT_TEMPERATURE
);
port ( 
  terminal NC : ELECTRICAL;
  terminal NB : ELECTRICAL;
  terminal NE : ELECTRICAL;
  terminal NS : ELECTRICAL           -- default: ELECTRICAL_ref
);
end component BJT;

-- JFET (SUM §3.4.5)

component JFET
generic ( 
  MDATA  : JFET_DATA ; -- model data
  AREA   : REAL       := 1.0;  -- area factor
  TEMP   : REAL       := AMBIENT_TEMPERATURE
);
port ( 
  terminal ND : ELECTRICAL;
  terminal NG : ELECTRICAL;
  terminal NS : ELECTRICAL
);
end component JFET;
-- MOSFET (SUM §3.4.7)

component MOSFET
  generic
    MDATA : MOSFET_DATA; -- model data
    L : REAL := REAL'LOW; -- channel width (in m)
    W : REAL := REAL'LOW; -- channel length (in m)
    AD : REAL := REAL'LOW; -- area of drain diffusion (in m^2)
    AS : REAL := REAL'LOW; -- area of source drain diffusion
    PD : REAL := 0.0; -- perimeter of drain junction
    PS : REAL := 0.0; -- perimeter of source junction
    NRD : REAL := 1.0; -- equivalent number of squares of drain diffusion
    NRS : REAL := 1.0; -- equivalent number of squares of source diffusion
    TEMP : REAL := AMBIENT_TEMPERATURE;
  end generic;
  port
    terminal nd : ELECTRICAL;
    terminal ng : ELECTRICAL;
    terminal ns : ELECTRICAL;
    terminal nb : ELECTRICAL;
  end port;
end component MOSFET;

-- MESFET (SUM §3.4.9)

component MESFET
  generic
    MDATA : MESFET_DATA; -- model data
    AREA : REAL := 1.0 -- area factor
  end generic;
  port
    terminal ND : ELECTRICAL;
    terminal NG : ELECTRICAL;
    terminal NS : ELECTRICAL;
    terminal NB : ELECTRICAL;
  end port;
end component MESFET;

end package SPICE_COMPONENTS;
Comments
B1. Comments of E. Christen concerning Version 0.4

Comments on the Proposed SPICE2VHD Model Library
================================================
Ernst Christen, Synopsys, Inc.

General Comments
----------------
The approach is essentially based on the paper
Ernst Christen, Kenneth Bakalar: "Library Development Using the
VHDL-AMS Language", in Jean Mermet (Ed.): "Electronic Chips &

There are some inconsistencies in referring to Spice versions in [2].
For example,
p.2: SPICE3 Version 3f3
p.4 et.al: SPICE3F
p.7 SPICE3F5

It is not clear from these specifications who provides what. My first
interpretation is that the packages define components, which means that
the interfaces of the models are specified. It is not so clear whether
VDA-FAT-AK30 will also provide a portable implementation of the corresponding
models. If this isn't done, then portability of a design is compromised
because each vendor will have their own implementation of say a mos level 3
model (including their specific bug fixes and enhancements). This cannot
be the intent of the effort.

states that everything should be in Kelvin. This should be consolidated.

Specific Comments
-----------------
[2] p.3: "SPICE-like current controlled voltage and current sources are
knowingly not considered because they require an access to internal data
objects of other models."
This is not necessarily the case. For example, it is possible
to create a voltage source that exports its current:

entity vdc is
  generic(dc: real);
  port(terminal P, N: electrical; quantity i_vdc: out real);
end entity vdc;

According to VHDL semantics a quantity port with mode out does not
have to be associated, which means that the source can be used like
a SPICE source. The quantity port must be associated if the current
is needed, for example to control the input of a current controlled
source.

[2] p.4: "A special implementation of these types is not required. It depends
on the implementation whether array or records are used."
I believe this is weak. A user has no way of defining a constant of
such a type in a portable way if he doesn't use the SET_* functions.
There is no way to report the value of such a constant in a portable
way.
I understand that the proposal is based on the fact that some
implementations do not support record types at this time. This should
not prevent VDA/FAT-AK30 to propose a correct approach using record
types, thereby forcing the deficient implementations to support record
types. At the last, the record type should be recommended, with an
array implementation accepted until record types have been implemented.

Having parameters depend on TSTEP or TSTOP is weak. I know that SPICE
does this, but an implementation of the SPICE library doesn't have to
replicate the bad choices of SPICE. To me it would be perferable if
either users were required to provide values for such parameters,
or if they were given defaults of REAL'HIGH.

[2] p.6 Complex Elements
Are there no polynomial controlled sources in SPICE 3f3?
I already mentioned current-controlled elements.

[2] p.8 constants row1 and row2
As we discussed in Lille, the expectation seems to be that the
SET_*_DATA function be overloaded, with one version being able to
parse a SPICE .model card. As mentioned in the General Comments
section above, the question is who provides this functionality,
and in what form. Additionally, in this example, row1 and row2 are
concatenated,, making the result a single string, but roq2 still
contains the SPICE continuation character "+". This means that
the string in that section reads "... VAF=50+ CJB=...", which
cannot be parsed without error.

[2] p.17 component semiconductor_resistor
Why is there this constant that is then used in the component?
Why not call the SET_* function directly in the component?

[2] p.17 resistor/semiconductor_resistor
Why isn't there a single resisto that supports both sets of arguments?
I believe this is how it's done in SPICE, using precedence rules.

I don't think that these initial conditions, and others for some
semiconductor devices can be supported due to the semantics of
VHDL-AMS.

[2] p.20 vpwl
I consider defining the waveform as a real_vector as weak. It forces
the implementation to verify that an even number of values has been
provided. A better approach that enforces this aspect is to define
the types
type xy_type is record x,y: real; end record;
type waveform is array(natural <>) of xy_type;
and then to define wave to be of teh waveform type.

[2] p.20 vsine
Is the existence of a default for freq an accident? It doesn't
seem to make sense.

[2] p.22 diode
Has two arguments commented, but the same arguments are functional for
bjt et.al.

[3] p.5 remark about exp
This remark should be explained better.

[3] p.6 package spice_fundamental_constants
Why this package? I don't see SPICE_TNOM being used anywhere, and
there is a (deferred) constant AMBIENT_TEMPERATURE in package
material_constants which is part of 1076.1.1. To me it doesn't make
sense to separate SPICE_TEMPERATURE from AMBIENT_TEMPERATURE.

References
---------
[2] "SPICE Components in VHDL-AMS", Proposal for Discussion, Version 0.4,
July 16, 2004
[3] VHDL-AMS Modelbibliothek SPICE2VHD, version 1.0, September 1, 2004
Changes in Version 0.6 as consequence of the comments

General Comments

- Reference to paper by E. Christen and K. Bakalar included
- Reference to Spice3f3
- The nominal temperature is given by the constant SPICE_TNOM in degree Kelvin

Specific Comments

- [1] p. 3: To be in accordance with the Spice interface and also the Verilog-AMS proposal this was not changed. May be this should be discussed later on.
- [2] p. 4: A recommendation to use record types was added in section 2.1. However for the user of the models it is without consequences whether arrays or records are used because he should apply the SET_*-functions to specify MDATA values in the models.
- [1] p. 5: TSTOP was replaced by REAL’HIGH. If possible TSTEP was replaced by 0.0 (for example to define rising and falling times). Otherwise a mapping of values to parameters is required during instantiation (see for instance VEXP, IEXP). The same was done in the case of 1/TSTEP (see VSINE, ISINE, VSFFM etc.).
- [1] p. 6: Polynomial controlled sources are not supported in SPICE3f3. General non-linear controlled sources are supported. But it seems to be too complicated to support them in VHDL-AMS.
- [1] p. 8: The old version 2 using strings to declare .model card information was removed.
- [1] p. 12/[1] p. 17: The models RESISTOR and SEMICONDUCTOR_RESISTOR were combined. In the same way the CAPACITOR model was extended. A problem might be that the basic simple RESISTOR and CAPACITOR model are a little bit overloaded. The parameter DEFAULT_CAPACITOR_VALUE cannot be specified using SET_CAPACITOR_DATA because the parameters CJ and CJSW must be defined in some way if the function is called (compare section 8.2 of the SPICE3f3 User’s Manual). REAL’LOW is not possible for these values. In the current implementation of the SPICE2VHD library this combination was not done.
- [1] p. 19: All the initial conditions in TLINE and the semiconductor models were removed.
- [1] p. 20 (vpwl): The declaration with the WAVE vector is in accordance with SPICE3f3 and also the Verilog-AMS proposal. That is why it was not changed. May be this should be discussed later on.
- [1] p. 20 (vsine): The frequency 0 was the consequence of TSTOP equal REAL’HIGH and the requirement of a default value equal 1/TSTOP. It was changed in that way that a value is required during instantiation. See also ISINE, VSFFM, and ISFFM and the comments in section 2.3.
- [1] p. 22: The arguments for the initial values were removed (see also TLINE comment).
- [2] p. 5: Accepted but not part of this document.
- [2] p. 6: This is a real problem. It was refused during the discussion about the Draft for “Standard VHDL Analog and Mixed-Signal Extensions – Packages for Multiple Energy Domain Support” to define the unit of AMBIENT_TEMPERATURE. Thus, it is not known whether in a special environment the value is given in degree Celsius or Kelvin. However, during the evaluation of a model the unit must be known. To avoid problems the constant SPICE_TEMPERATURE was introduced in the test implementation. The constant defines the ambient temperature in Kelvin. Nevertheless, in this...
document always AMBIENT_TEMPERATURE is used. By the way Verilog-AMS function $temperature has to provide the value in degree Kelvin.