



Simul'Elec, a Delphi written simulator for power Electrical Engineering, using VHDL-AMS modeling



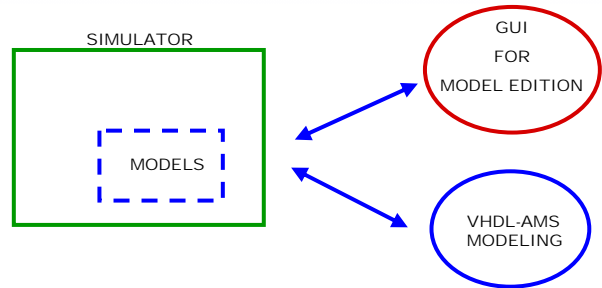
María Azurmendi, Fabien Legrand, Fernando Martin, Luis Fontán, Jean-Jaques Charlot, Nadine Couture
CEIT and TECNUN (University of Navarra), Spain. {mazurmendi, fmartin, fontan}@ceit.es
ALGO'TECH INFORMATIQUE, FRANCE. contact@algotech.fr
Jean-jacques.charlot@wanadoo.fr, n.couture@estia.fr

Abstract

Simul'Elec is an electrical simulator that is part of electrical CAD software. It is designed with the aim of aiding CAD software user, who are not very familiar with simulation tools. This friendly simulator, written in Delphi, integrates a graphical user interface to generate VHDL-AMS models and a compiler to simulate them.

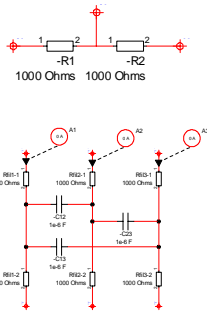
I - Introduction

- Most of the Electrical Simulators are developed for system design and require a good working knowledge of the behavior of each modeled component and operation of the simulator.
- System maintenance and evolution could be successfully improved by electrical simulation. For this, the simulator must be able to provide results with no need of modeling skills or advanced knowledge of the operation of the simulator calculation core.
- Our solution is based on a user friendly graphical interface to produce a VHDL-AMS model, thereby enabling the use of the capabilities of this standard modeling language to define more complex behaviors.

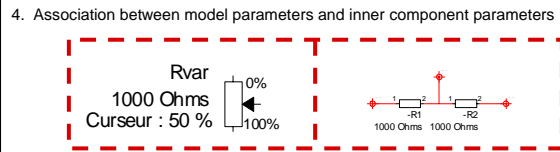


II-Graphical User Interface for Model Edition

- One of the major improvements about simulation is allowing user to create custom models, including behavioral modeling.
- Steps of graphical definition:



1. Drawing the model circuit and its external connections.
2. Voltage and current characteristic values, for display purpose at different simulation time.



3. Model parameters.
4. Association between model parameters and inner component parameters



5. Model backup

VHDL-AMS MODEL

III-VHDL-AMS Modeling

```

LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
USE IEEE.MATH_REAL.ALL;
ENTITY ParamResistor IS
    GENERIC (
        --@Name--@Description--Total Resistance
        --@Unit Type--@MinValueRange--@MaxValue--@
        --@MinValue--@MaxValue--@Unit--@Position--@
        --@PossibleValues--@Unit--@Value--1000
        Rval: REAL := 1000;
        --@Name--CurPos@Description--Cursor position percentage
        --@Unit Type--@MinValueRange--@MaxValue--@
        --@MinValue--@MaxValue--@Unit--@Position--@
        --@PossibleValues--@Unit--@Value--50
        CurPos: REAL := 50;
    );
    PORTS (
        TERM1: EXT1; EXT2: EXT3 ELECTRICAL;
    );
END ENTITY ParamResistor;
ARCHITECTURE Sim OF ParamResistor IS
    --parameter values for resistance calculation
    QUANTITY Comp_R1_Param_Resistance: REAL;
    QUANTITY Comp_R2_Param_Resistance: REAL;
BEGIN
    Comp_R1_Param_Resistance <= Rval*(CurPos/100);
    Comp_R2_Param_Resistance <= Rval*(1-CurPos/100);
R1: ENTITY Component_alimentation_RESISTANCE(Beh)
    GENERIC MAP (Resistance => Comp_R1_Param_Resistance)
    PORT MAP (EXT1 => EXT1, EXT2 => EXT2);
R2: ENTITY Component_alimentation_RESISTANCE(Beh)
    GENERIC MAP (Resistance => Comp_R2_Param_Resistance)
    PORT MAP (EXT1 => EXT2, EXT2 => EXT3);
END ARCHITECTURE Sim;
    
```

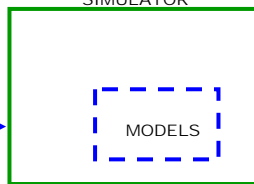
- In order to enable the user to change some features of the electrical models easily, the models are described in an intuitive and standard textual modeling language like VHDL-AMS.

VHDL-AMS EDITOR



- The source code of the model is shown in the VHDL-AMS Editor, where the user can edit some parts of the Architecture body.

SIMULATOR



- Thanks to the compiler developed, the VHDL-AMS code edited by the user is lexically, syntactically and semantically checked. Finally a Delphi model is generated, a code understandable by the Simulator.

IV- Conclusion

- Simul'elec users define and edit electrical models from a friendly Graphical User Interface and VHDL-AMS modeling scheme integrated in this electrical simulator.

- The multi-technological capabilities provided by VHDL-AMS language make possible the representation of non-electrical behavior such as electromechanical devices and thermal phenomena. This new abstraction will be integrated into the current VHDL-AMS scheme.

