ModCoupler-VHDL

User's Guide

Powersim Inc.

ModCoupler-VHDL User's Guide

Version 1.0

Release 1.0

May 2011

Disclaimer

Powersim Inc. ("Powersim") and the Carlos III University of Madrid make no representation or warranty with respect to the adequacy or accuracy of this documentation or the software which it describes. In no event will Powersim and the Carlos III University of Madrid or its direct or indirect suppliers be liable for any damages whatsoever including, but not limited to, direct, indirect, incidental, or consequential damages of any character including, without limitation, loss of business profits, data, business information, or any and all other commercial damages or losses, or for any damages in excess of the list price for the license to the software and documentation.

Powersim Inc.

Email: support@powersimtech.com
powersimtech.com

1 Introduction

ModCoupler-VHDL is a communication link between the simulation software ModelSim® and PSIM®. By means of ModCoupler-VHDL, the co-simulation of the completed power electronics equipment can be performed. The digital control algorithm, described in VHDL, will be simulated in ModelSim® and the power stage will be simulated in PSIM.

On the PSIM side, the user must include the ModCoupler-VHDL block in his schematic. On the ModelSim side there, no changes are needed. From PSIM´s schematic, the user will provide information to ModCoupler-VHDL module, time-step, VHDL clk signal frequency and input – output signals information, required to configure the co-simulation. Fig. 1 shows the basic co-simulation structure.

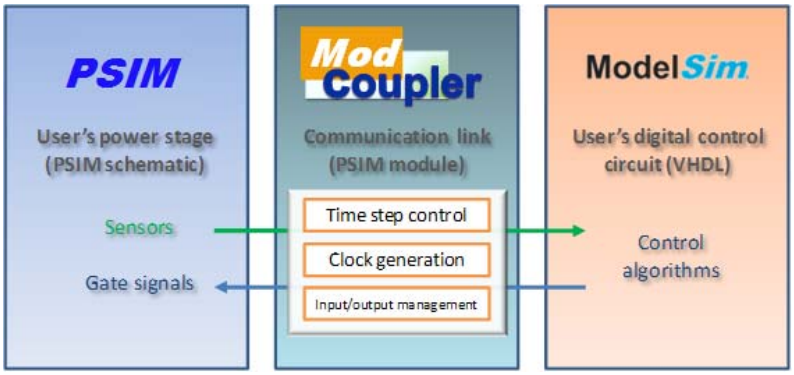

Fig. 1: Co-simulation architecture.

At every simulation instant both simulators are stopped. The input signal values of the ModCoupler module are forwarded to the digital circuit, which is in charge of the control algorithms. Once the ModelSim´s calculation is finished, the VHDL output values are sent back to close the loop. After that, each simulator runs a new simulation step and the cycle is repeated. Note that one PSIM simulation step correspond to several ModelSim simulation steps.

This guide describes how to build a co-simulation environment from scratch. It is assumed that the user provides a PSIM schematic for the analog simulation and a VHDL description for the digital simulation.

2 ModCoupler-VHDL block configuration

In your PSIM schematic file (create a new one if it is needed) add a ModCoupler-VHDL block (placed on the *Control* submenu of the *Elements* menu) as show in Fig. 2.

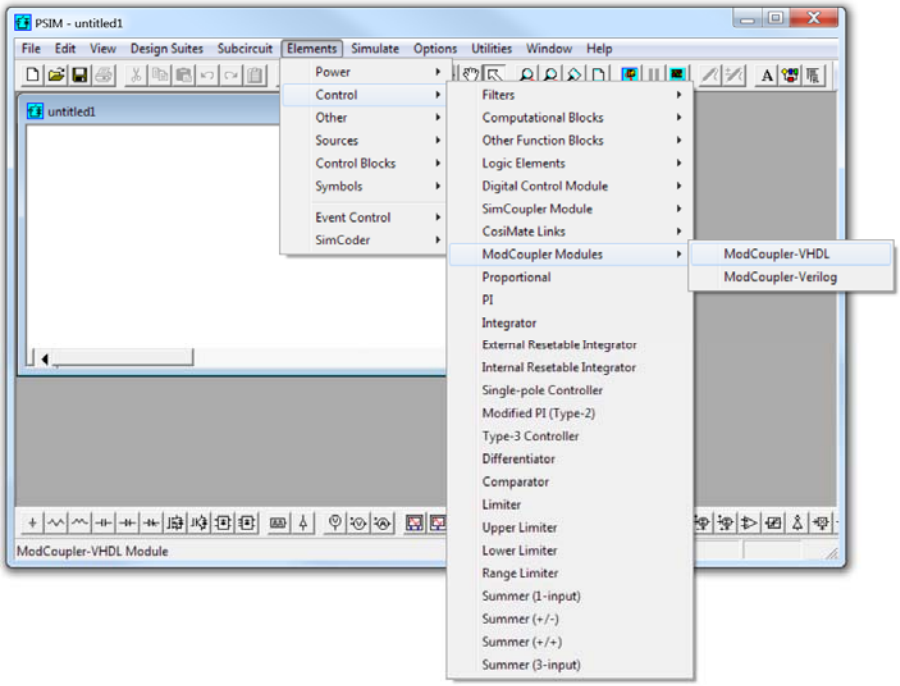

Fig. 2: ModCoupler-VHDL module menu.

In Fig. 3 the main dialogue window of the ModCoupler-VHDL module is shown.

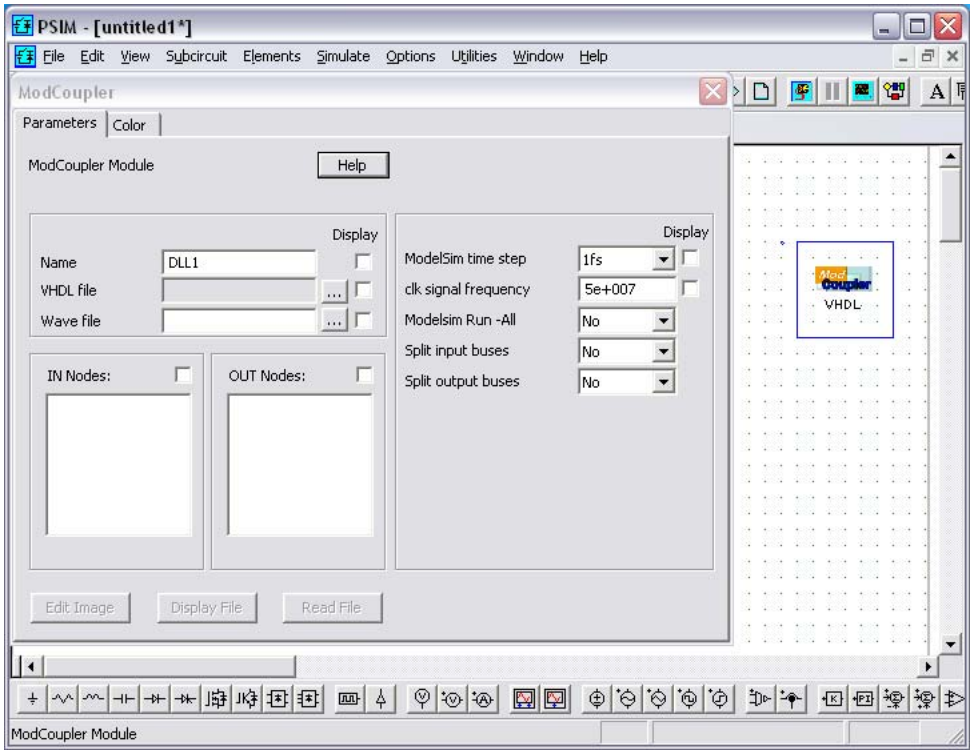

Fig. 3: ModCoupler-VHDL module dialogue window.

The different parameters are explained below:

- *VHDL file:*The .vhd top entity file. After this file is selected, the IN/OUT nodes lists will be created.
- *Wave file:* The file with the signals to be displayed in ModelSim waveform window.
- *ModelSim Time Step:* The ModelSim simulation time step. This value must be smaller than the PSIM time step of the Simulation Control
- *Clk signal frequency:* The frequency of the ModelSim clk signal. Notice that the clk signal is not processed as an input, since the period is usually smaller than the PSIM time step.
- *ModelSim Run All (Yes/No):* Allows to start the ModelSim simulation without pressing the "Run –All" button. It is recommended to set that parameter to "No" the first time the simulation is run in order to select the signals to be displayed in ModelSim waveform window.
- *Split input buses (Yes/No):* Allows splitting an input std_logic_vector signal in its different bits.
- *Split output buses (Yes/No):* Allows splitting an output std_logic_vector signal in its different bits.

3 Compatible data types

ModCoupler-VHDL handles the following type of signals:

3.1 bit

When used as input signal, ModCoupler-VHDL will set the corresponding ModelSim signal to logic '0' when reads a value of 0, otherwise the signal will be set to logic '1'.
When used as output signal, ModCoupler-VHDL will set the corresponding PSIM output node to 0 when reads a logic value of '0', otherwise the node will be set to 1.

3.2 std_logic

With the std_logic type only two states are taken into account, the logic '0' and the logic '1'.
When used as input signal, ModCoupler-VHDL will set the corresponding ModelSim signal to logic '0' when reads a value of 0, otherwise the signal will be set to logic '1'.
When used as output signal, ModCoupler-VHDL will set the corresponding PSIM output node to 0 when reads a logic value of '0', otherwise the node will be set to 1.

3.3 real

With this kind of signal, the PSIM values are directly sent to ModelSim and vice versa.

3.4 integer

When used as input signal, ModCoupler-VHDL will round the PSIM input node value truncating the decimal part and after that will set the corresponding ModelSim signal to the rounded value.

3.5 bit_vector, std_logic_vector

In these both cases there are two different modes of operation:
When the *Split input buses* parameter is set to "No", the bit vector will be treated in the PSIM side as an integer and ModCoupler-VHDL will set the first element of the array as the least significant bit (LSB) of the integer value, the second as the second LSB, etc.

When the *Split input buses* parameter is set to “Yes” an input node is created in PSIM for each of the signal bits.

4 Compilation of VHDL files

ModCoupler-VHDL needs a compiled model of the VHDL design in order to operate. Also, the “work” ModelSim library must be located in the working directory (it can be done easily moving the *work* folder to the directory that contains the schematic file).

The proposed compilation method uses a Windows® batch file, although this process should be performed by the ModCoupler-VHDL dialog. A batch file includes commands to be executed. Example batch files (compile.bat) can be found in the different examples directories. ModelSim applications *vcom* and *vlib* are used, so the path to both of them must be in the environment variable PATH.

NOTE: If any change is made on any VHDL file, the model must be recompiled.

5 Simulation

The last step is running the simulation. Set the configuration *ModelSim Run All* parameter to “No” the first time the simulation is run, so ModelSim opens without actually running the simulation. In this state, the user can select the appropriate signals to display and save them to the wave file (*wave.do* by defect).

Start the simulation pressing the PSIM “Run simulation engine” button. After a few seconds, a ModelSim window will appear. Once the user has selected the signals to view, the simulation is started by pressing the “Run –All” button in ModelSim.

In next simulations, the *ModelSim Run All* parameter can be set to “Yes” to start ModelSim and run the simulation automatically.

If a new simulation with the same VHDL model is needed (for example, after editing a VHDL file and recompiling or after a schematic change), press the ModelSim “Restart” button before pressing the PSIM “Run simulation engine” button again (closing ModelSim window is not required).