ModCoupler Tutorial

Grid connected three-phase voltage source inverter

The ModCoupler[1] provides interface for co-simulation between PSIM® and the software Modelsim®.

This tutorial is intended to guide you, step by step, to perform a co-simulation PSIM® – Modelsim® example based on the closed loop operation of a three phase inverter connected to the grid.

1.1 ModCoupler block configuration.

Before configuring the ModCoupler block, the PSIM® schematic will be created.

In this example PSIM® is in charge of the inverter power simulation while ModelSim® is doing the control part. The final schematic is included in the working directory (Example1) and its name is Inverter.psimsch but, in this example, the complete communication process between PSIM® and ModelSim® is explained. So, we will start with a three-phase power inverter connected to the grid (Fig. 1).
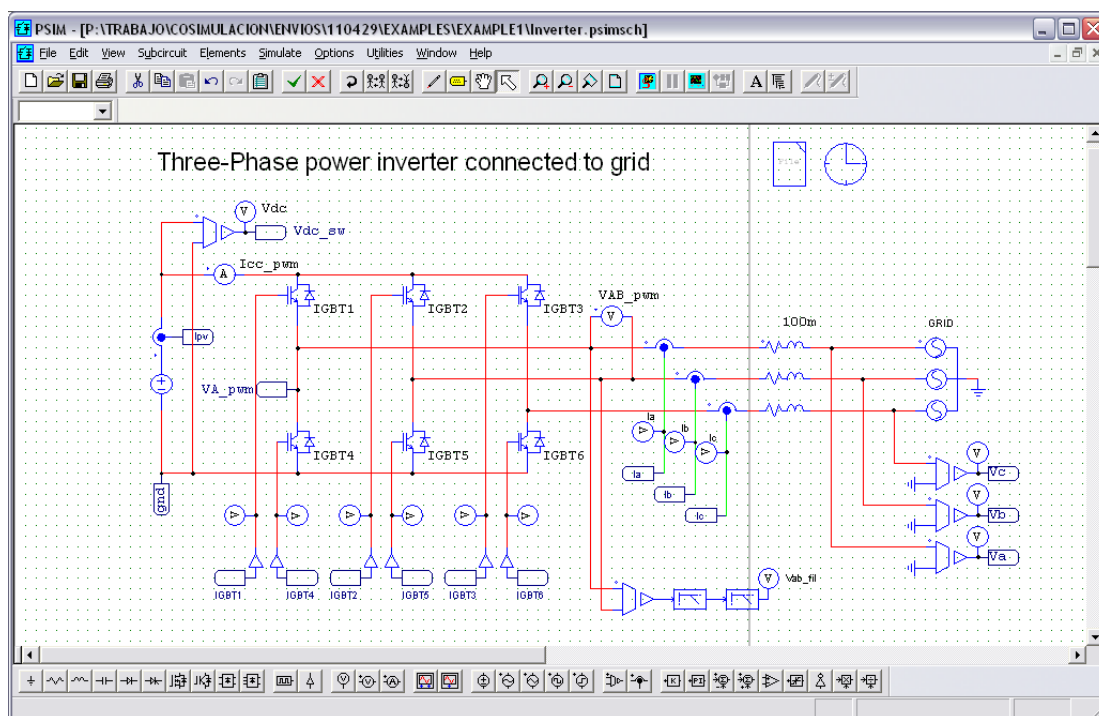

Fig. 1: PSIM´s schematic of the power stage of a grid connected three-phase inverter.

To make this design functional it's only necessary to give PSIM® the IGBT's pulses (IGBT1-IGBT6 in the figure). These pulses will be calculated by means of the ModelSim® simulator. For this purpose we'll add a ModCoupler block and select Top_Inversor.vhd (located in the *vhdl* folder of this example) as VHDL file (Fig. 2).The IN/OUT Nodes lists will be created.

The outputs will be obviously connected to the IGBT's gates in order to commutate them. The 12 different inputs are the following:

- Reset: The reset signal of the digital design.
- Ia, Ib and Ic: The line currents.
- Va, Vb and Vc: The line voltages.

---

[1] ModCoupler is copyright in 2011 by Carlos III University of Madrid, GSEP Power Electronics Systems Group and Microelectronic Design and Applications Group, Spain.

- Vdc_sw: The input voltage.
- W: The line angular frequency in rad/s.
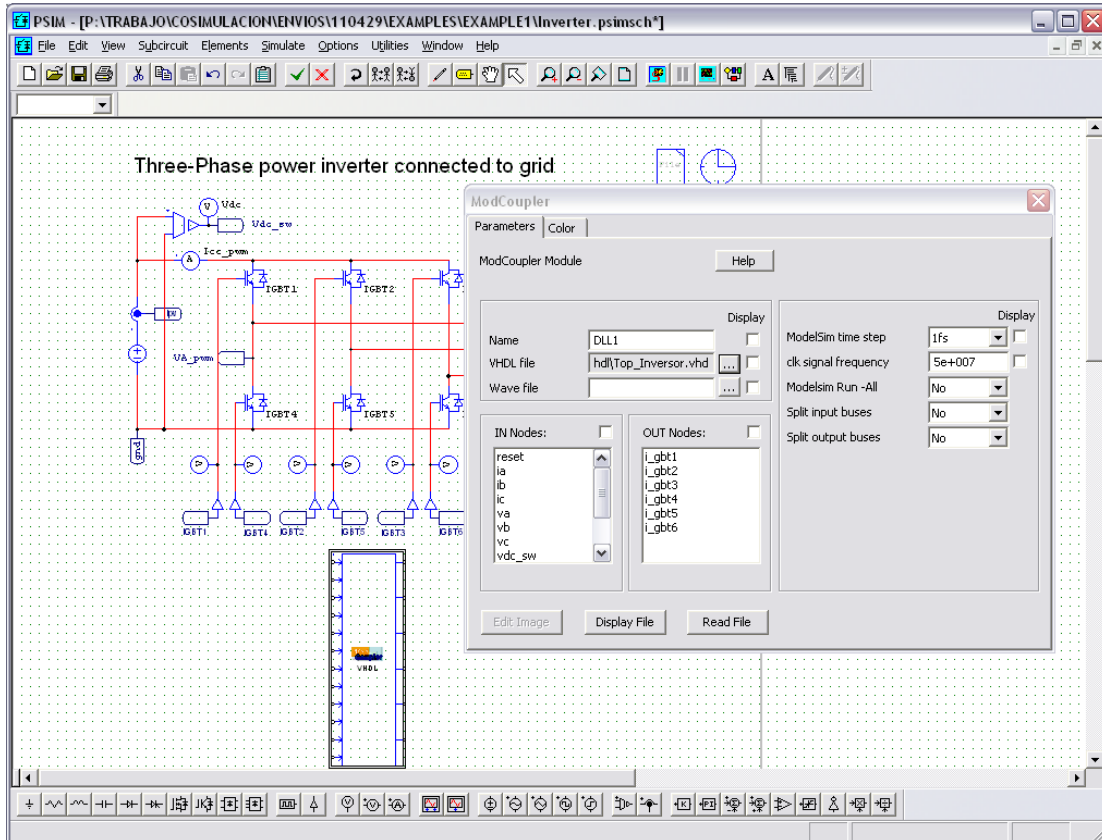- Pref and Qref: The active and reactive power references.
- L: The line inductance.

Fig. 2: ModCoupler Block dialog window.

In this example, the type of all the input signals is *real* except for the *reset* that is a *std_logic* signal. The use of the real type eases the connection process as the PSIM® and the ModelSim® values are the same. However, this type of signal cannot be synthesized so it is not recommended for a design that will be tested in a real system.

Next, the ModCoupler Block parameters will be set with the following values:

Wave file: Point to the wave.do file placed on the example folder.

ModelSim time step: 100ns

Clk signal frequency: 125000

ModelSim Run All: Yes

Split input buses: No

Split output buses: No

In Fig. 3 the final the ModCoupler Block is shown. Also, the Pref, Qref, w and reset signals are created using voltage sources.
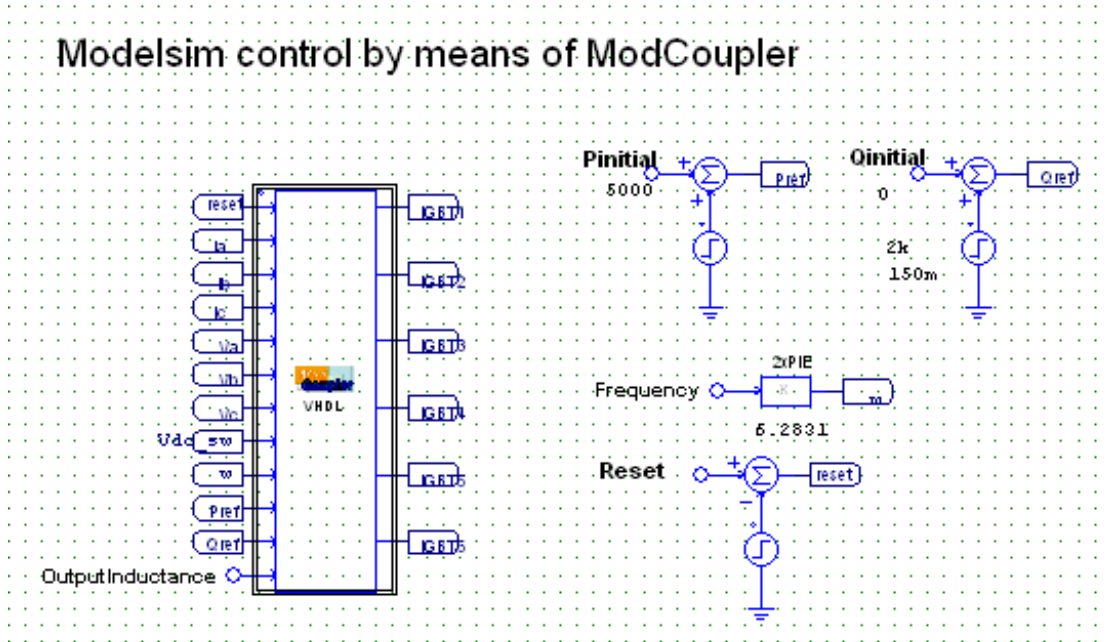

Fig. 3: Final control schematic.

1.2 Compilation of VHDL files

The compiled version of the model is included in the work folder included in the example main directory, anyway, a batch file is given to re-compile the design if it is needed.

This batch file is located in the subfolder "vhdl" and it is called compile.bat. Double click to execute them. In this batch file, the ModelSim® applications *vcom* and *vlib* are used, so the path to both of them must be in the environment variable PATH.

This .bat file also moves the "work" subfolder to the main folder of the example ("ModCoupler Tutorial 1"). If other compilation method have been used (e.g. using de command prompt or the ModelSim® IDE), the created "work" folder must be moved manually to the work directory ("ModCoupler Tutorial 1" in this example).

1.3 Simulation

The last step is run the simulation. Start it by pressing the PSIM® "Run simulation engine" button. At this point, ModCoupler creates a VHDL file called ModCouplerTemporaryFile.vhd in the vhdl directory and compiles it. After a few seconds, a ModelSim® window will appear with the compiled model. As the RunAll parameter was set to "Yes", the ModelSim® simulation will start immediately.

NOTE: ModCoupler uses the ModelSim® applications *vcom* and *vsim*, so the path to both of them must be in the environment variable PATH