ModCoupler Tutorial

DC/DC converter

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

In this tutorial, the closed loop control of a synchronous buck DC/DC converter is simulated. The power stage is simulated in PSIM® and the controller is described in VHDL and simulated with ModelSim®. The VHDL design has been developed from a "prototype" point of view, i.e., taking into account the format signals that are used in a physical implementation.

1.1 ModCoupler block configuration.

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

In this example PSIM® simulate the power stage of a DC/DC converter, as well as the analog-to-digital converter. ModelSim® simulates the control loop implemented in VHDL code. The final schematic is included in the working directory (ModCoupler Tutorial 2) and its name is SynchronousBuck.psimsch. However in the next paragraphs, the process to include in the schematic the blocks to communicate PSIM® and ModelSim® will be explained in order to show to the user how can connect its own design.

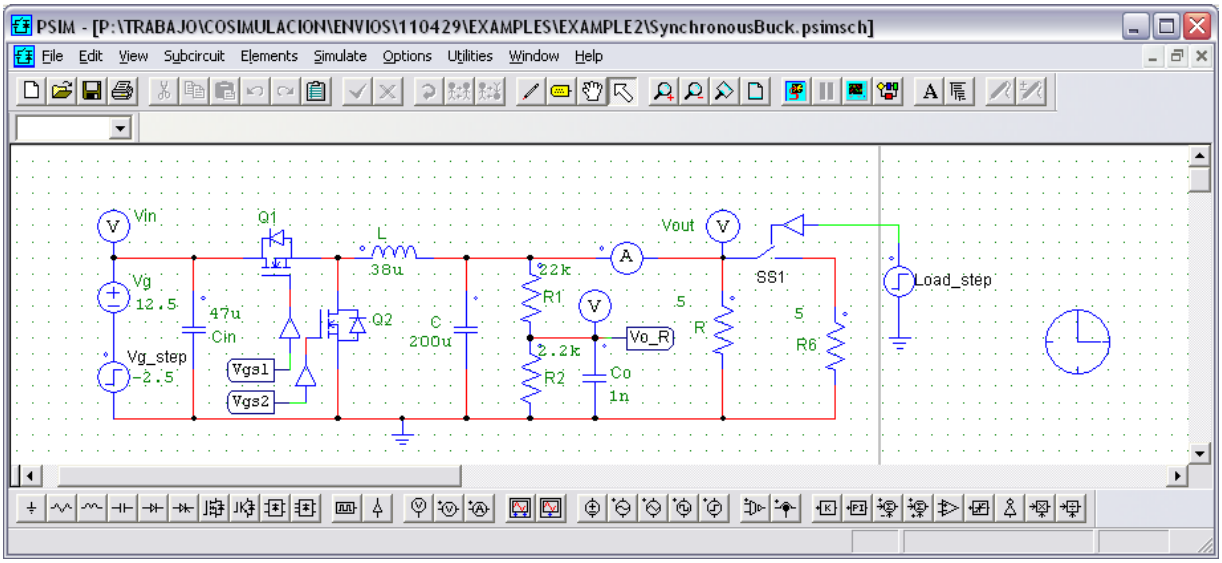

Fig. 1: Synchronous buck converter. Additional elements have been added to simulate input voltage steps and load steps.

In Fig. 1 the power stage is shown. Additional elements have been added to simulate voltage steps (Vg_step) and to simulate load steps (R6, SS1 and the voltage source Load_step). The control loop is fed with the output voltage signal Vo_R adapted with a resistor divider (R1 and R2), and it generates the control pulses Vgs1 and Vgs2 of the switches Q1 and Q2 respectively. The control loop has been designed in VHDL, which is simulated with ModelSim®. In order to complete the schematic the ModCoupler block to communicate with ModelSim® have to be inserted, as well as an A/D converter PSIM® block simulating the conversion of the output voltage signal to digital values understandable by the VHDL module.

First, an 8 bit A/D converter PSIM block is inserted (Fig. 2) to convert the analog signal Vo_R to digital values understandable by the digital controller. The VHDL controller generates also the control signal of the A/D converter.

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

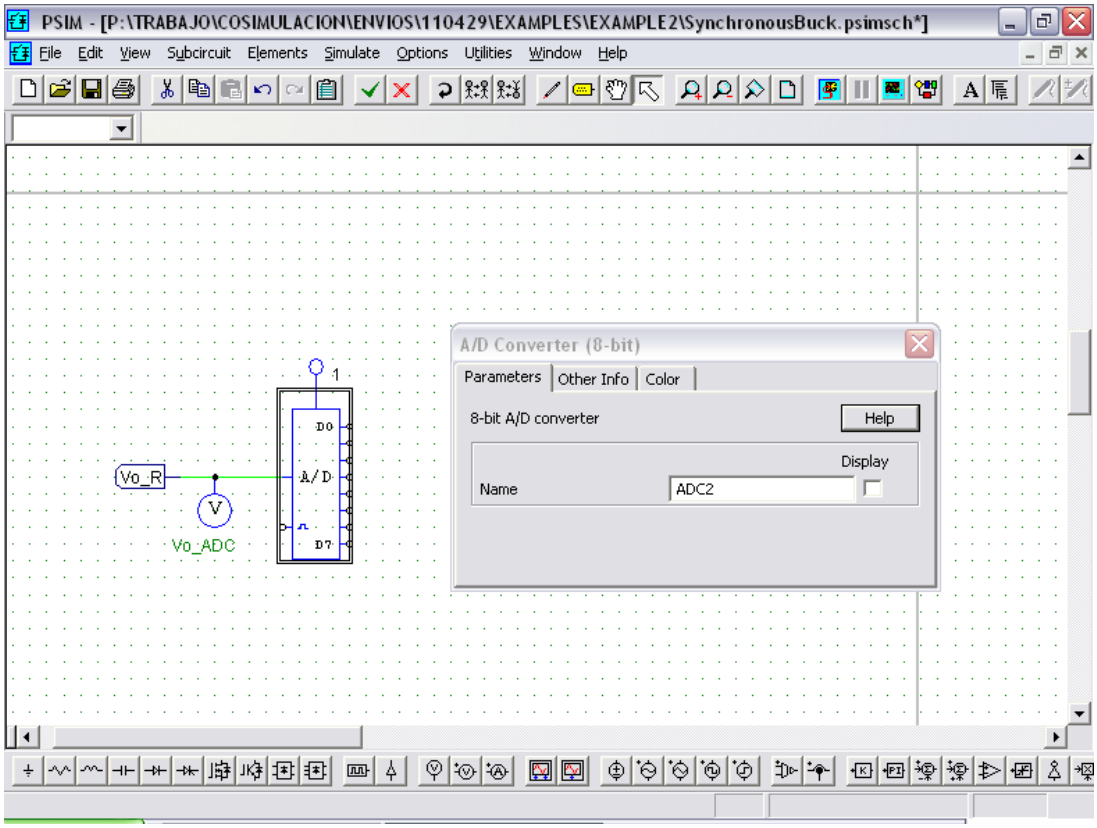

Fig. 2 A/D converter block.

A ModCoupler Block is inserted and the top_controller.vhd file (located in the *vhdl* folder of this example) is selected as VHDL file, and the *Split input buses* and *Split output buses* parameters are set to "Yes" (Fig. 3).The IN/OUT Nodes lists will be created.

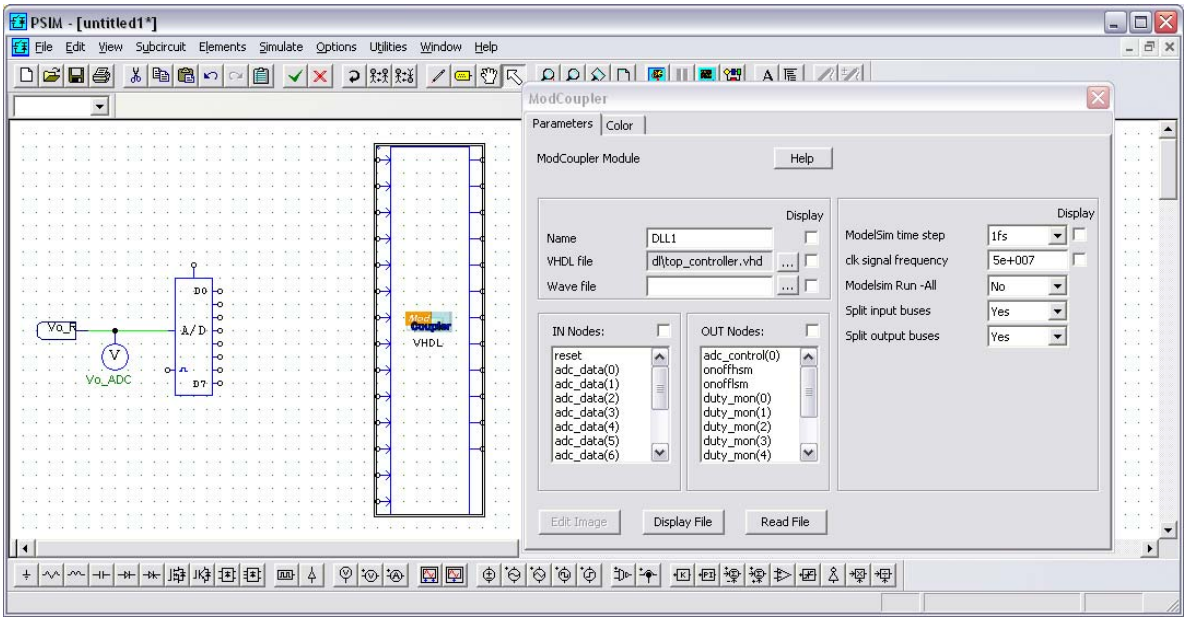

Fig. 3 ModCoupler Block dialogue window.

The VHDL controller provides the duty cycle value through 9 output bits. In order to assess the duty cycle value, a 10 bits D/A converter PSIM® block is used, as shown in Fig. 4.

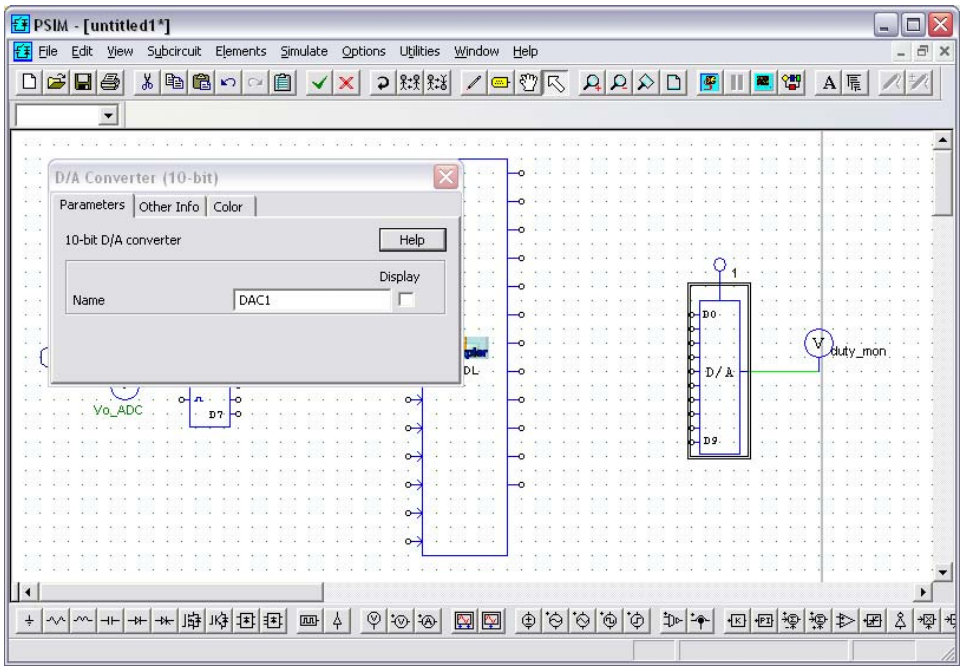

Fig. 4: D/A converter for duty cycle monitoring.

The three control blocks to be inserted in this example in PSIM® have been described. The next step is to interconnect the three control blocks with among them and with the power stage. To ease the task of connect each input/output node with the corresponding signal, it is recommended to check the IN/OUT Nodes checkboxes placed in the ModCoupler dialogue window. The connection of the three control blocks is shown in Fig. 5.

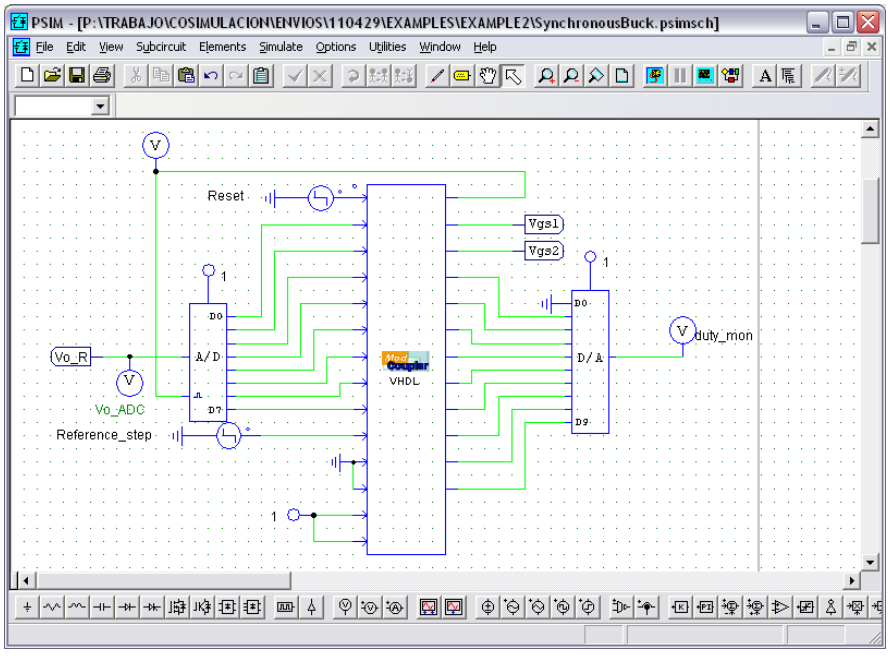

Fig. 5 Connection of the three blocks inserted on PSIM® to be connected with the power stage.

The final and complete schematic is shown in Fig. 6.

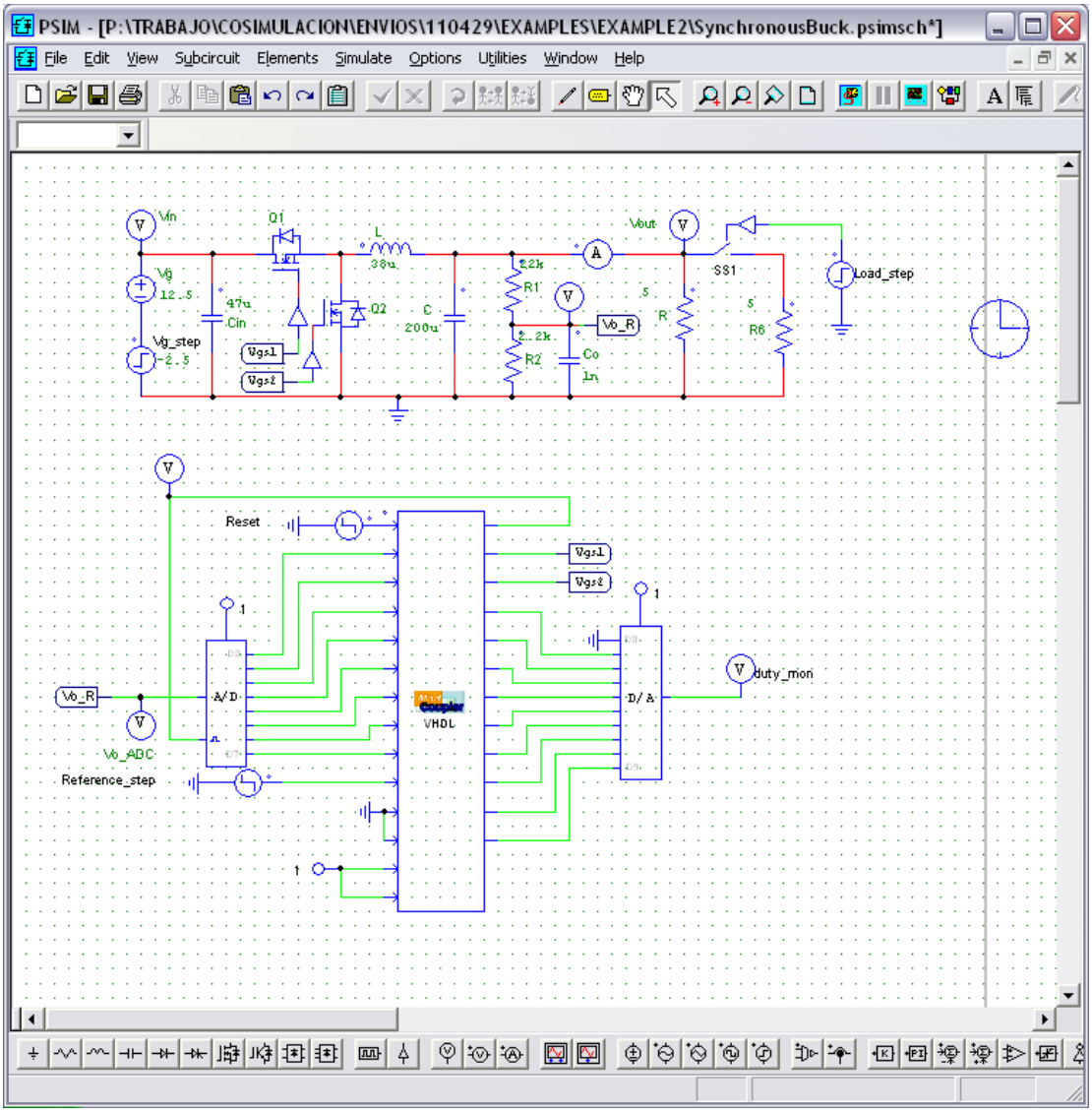

Fig. 6 Final schematic.

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*: 10ns
- *Clk signal frequency*: 5E7
- *ModelSim Run All*: No
- *Split input buses*: Yes
- *Split output buses*: Yes

Splitting the input and output buses allow to use a different input/output for each of the std_logic_vector signal bits.

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 2). 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 2” 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 “No”, the ModelSim® simulation waits until user press “Run –all” button.

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