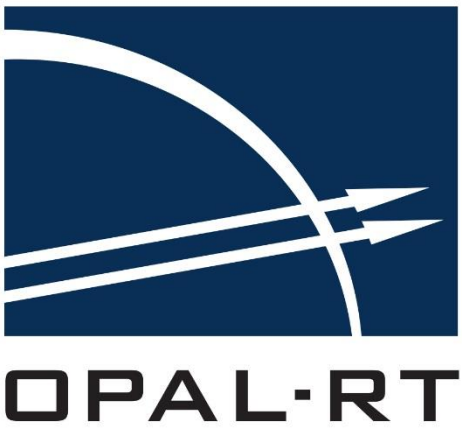


eHS User Guide

www.OPAL-RT.com

Table of Contents

1. INTRODUCTION 5

1.1 The Organization of This Guide 5

1.2 Conventions 5

1.3 About the OPAL-RT eHS From the eFPGAsim Toolbox 5

1.4 Key Features 5

1.4.1 Configurability 5

1.4.2 Performance 6

1.4.3 Compatibility 6

1.4.4 Scalability 6

1.4.5 Test Scenarios 6

1.5 Specifications 7

1.5.1 eHS Gen3 7

1.5.2 eHSx64 (Gen2 - Deprecated) 8

1.5.3 eHS (Dual eHS) (Gen1 - Deprecated) 8

1.6 Intended Audience and Required Skills and Knowledge 8

1.7 External Tools 8

1.7.1 Simulink 8

1.7.2 SimPowerSystems and PLECS Simulink toolboxes 8

1.7.3 PSIM 9

1.7.4 NI Multisim 9

2. REQUIREMENTS 10

2.1 Software Requirements 10

2.2 Installation Notes 12

3. BUILDING MODELS WITH THE EHS SOLVER 13

3.1 Circuit-Under-Test Design Using the SimPowerSystems Simulink Toolbox 13

3.1.1 Circuit Design Using the SimPowerSystems Libraries 13

3.1.2 Supported Blocks 13

3.1.3 Component Naming 16

3.2 Circuit-Under-Test Design Using the PSIM Toolbox 18

3.2.1 Circuit Design Using the PSIM Libraries 18

3.2.2 Supported Blocks 18

3.2.3 Component Naming 19

3.3 Circuit-Under-Test Design Using the PLECS Blockset Simulink toolbox 21

3.3.1 Circuit Design Using the PLECS Libraries 21

3.3.2 Supported Components 21

3.3.3 Example of Circuit Design Using PLECS 22

3.4 Inserting the Circuit into an RT-LAB Model 23

3.4.1 eHS Blocks 23

3.4.1.1 The eHSx32 Gen3 Block for the OP4200 23

3.4.1.2 The eHS Gen3 Block 26

3.4.2 Signal Types 26
3.4.3 Offline Simulation 27
3.4.4 Understanding Sample Times 27
3.5 Interconnecting Multiple eHS Solvers 28
3.6 Using the Scenario Feature 30
3.6.1 What is a Scenario? 30
3.6.2 Writing the XLS Scenario File 30
3.6.2.1 Getting Started 31
3.6.2.2 Filling the Component Line 31
3.6.2.3 Filling the Scenario Column 31
3.6.2.4 Default Scenario Line 31
3.6.2.5 Removing Lines/Columns 31
3.6.2.6 Filling the Table 31
3.6.2.7 Number of Modifications Allowed by Scenario 31
3.6.2.8 Non-Existing Component 31
3.6.2.9 Non-Existing Scenario Declaration 31
3.6.2.10 Verifying That Scenarios Have Been Properly Generated 32
3.7 The Loss Compensation Algorithm (LCA) Feature (eHS Gen3 Only) 33
3.7.1 What is the LCA? 33
3.7.2 Recognizing the Gen3 block 33
3.7.3 Supported Topologies 34
3.7.4 How to Use the LCA Feature (SimPowerSystems workflow) 35
3.7.4.1 Using LCA With a Two-Level Arm Converter 35
3.7.4.2 Using the LCA Feature on a Three-Level NPC Arm Converter 37
3.8 Example Models and Templates 39

1. INTRODUCTION

1.1 THE ORGANIZATION OF THIS GUIDE

This document is the user guide, which covers the following topics:

- **Introduction** - Provides an introduction to simulation and the principles behind the use of the eHS solver.
- **Requirements** - Software requirements for the eFPGAsim toolbox, including the eHS solver.
- **Building models with the eHS solver** - Describes the procedure to develop an RT-LAB model with the eHS solver.

1.2 CONVENTIONS

OPAL-RT guides use the following conventions:

Table 1: General and Typographical Conventions

CONVENTION	DESCRIPTION
Bold	User interface elements, text that must be typed exactly as shown.
Note:	Emphasizes or supplements parts of the text. You can disregard the information in a note and still complete a task.
Warning:	Describes an action that must be avoided or followed to obtain desired results.
Recommendation:	Describes an action that you may or may not follow and still complete a task.
Code	Sample code
Italics	References work titles
Blue Text	Cross-references (internal or external) or hypertext links

1.3 ABOUT THE OPAL-RT EHS FROM THE EFPGASIM TOOLBOX

The OPAL-RT electric Hardware Solver (eHS) is a powerful floating-point solver developed by OPAL-RT that enables users to simulate an electric circuit on an FPGA automatically, without having to write the mathematical equations. It combines the simplicity of building electric circuit models using the SimPowerSystems Toolbox, PSIM, the PLECS Blockset, or NI Multisim software with the strength of OPAL-RT FPGA-based simulators to solve the currents and voltages within the circuit in real-time, with a sample time below 1µs.

The eHS solver uses the modified nodal analysis. It solves a conductance matrix to find the voltage at each node of the circuit, and the current in each branch. The conductance matrix of the circuit is made independent of the switch control signals through the implementation of the Pejovic[1] method representing the switch impedance. With this method, a conducting switch is represented as an inductance and an open switch is represented as a capacitance so the conductance matrix does not change during the simulation.

1.4 KEY FEATURES

1.4.1 CONFIGURABILITY

The conductance matrix does not need to be re-computed since switches are represented using the Pejovic method. All the matrices needed for solving the system are loaded onto the FPGA engine when the simulation is initiated. Reconfiguring the engine with specific firmware for each application is not

[1] Pejovic, P.; Maksimovic, D.; , "A new algorithm for simulation of power electronic systems using piecewise-linear device models," *Power Electronics, IEEE Transactions on* , vol.10, no.3, pp.340-348, May 1995

necessary. This feature makes running real-time simulations of electrical systems at high sampling rates as easy as any RT-LAB simulation.

1.4.2 PERFORMANCE

The sample time of the electrical system solved by the eHS solver ranges from 160ns to 4µs, depending on the circuit complexity.

1.4.3 COMPATIBILITY

eHS solvers provide a flexible environment that allows the user to design the simulated circuit in various netlist editors such as SimPowerSystems and PLECS Simulink toolboxes, Powersim PSIM, and NI Multisim software.

1.4.4 SCALABILITY

eHS exists in various form factors to accommodate low to high-end FPGA-based platforms. It is capable of simulating a Boost circuit with time step as low as 200ns, as well as a micro-grid system with outstanding resolution. Depending on the use case, it is possible to interconnect several eHS cores using a multi-FPGA system with low latency SFP 5Gbps links between chassis.

1.4.5 TEST SCENARIOS

It is possible to modify the component values during simulation to apply load variations and faults using the scenario feature. Please check the scenario feature description of this manual.

1.5 SPECIFICATIONS

1.5.1 EHS GEN3

The circuit is designed using blocks in supported netlist editors. A limited number of elements, chosen from a specific list, can be included in each circuit. The maximum number of circuits that can be simulated in one real-time model depends on the firmware installed onto the solver hardware and the number of available eHS licenses installed onto the simulation system. As opposed to previous generations, the 3rd Generation of eHS comes in various sizes to respond to different needs in terms of FPGA size, simulation complexity, and compatibility. Table 1 describes the existing eHS gen3 cores followed by their associated specifications:

Features	eHSx16	eHSx32	eHSx64	eHSx128
Targeted platforms	cRIO 7068 (Zynq 7020)	OP4200 (Zynq 7030)	OP4510 (Kintex7 325t)	OP5707 (Virtex 7 485t)
Number of inputs	16	32	32	128
Number of outputs	16	32	32	128
Number of switches	24	48	72	144
LCA capability[2]	Yes	Yes	Yes	Yes
Maximum number of states[3]	60	100	150	300
Number of resistors	Unlimited			
Switches type supported	IGBT/Diode, Diode, Breaker, Thyristor, Ideal Switch			
Non-switching devices supported	Resistor, Inductor, Capacitor, Ideal Transformer, Mutual inductance, PI Line			
Calculation power	6.4 GFLOPS	12.8 GFLOPS	25.6 GFLOPS	51.2 GFLOPS
Maximum number of test scenarios[4]	Up to 512 scenarios			
Circuit editors compatible	SimPowerSystems Simulink toolbox, PSIM, PLECS Blockset Simulink Toolbox, NI Multisim			

Table 1 -eHS Gen3 specification table

[2] LCA stands for Loss Compensation Algorithm. This feature optimizes losses for standard topologies such as the two-level and the three-level NPC arm converters.

[3] Estimated values. The maximum number of states depends on the number of inputs and outputs that needs to be computed as well. There is no hard coded limit. If the time step required exceeds the solver's limit (2.56us), a compilation error will occur due to overpassing the circuit size limit.

[4] The number of scenario available for a given circuit depends on the circuit complexity. Scenarios are not currently supported on the OP4200 target.

1.5.2 eHSx64 (Gen2 - Deprecated)

The circuit is designed using blocks from the SimPowerSystems library. A limited number of elements, chosen from a specific list, can be included in each circuit. The maximum number of circuits that can be simulated in one real-time model depends on the firmware installed in the solver hardware and the number of available eHS licenses in the simulation system. The number of elements in each circuit is subject to the following limitations:

- S = 1..64 switching elements
- U = 1..32 current or voltage inputs
- Y = 1..32 current or voltage measurements
- NSD = 0..150 non-switching devices. The number of non-switching devices includes L and C. "LC" and "RLC" branches each count for two non-switching devices.
- The number of resistors is not limited.
- Scenario support (up to 127).

1.5.3 eHS (Dual eHS) (Gen1 - Deprecated)

The circuit is designed using blocks from the SimPowerSystems library. A limited number of elements, chosen from a specific list, can be included in each circuit. The maximum number of circuits that can be simulated in one real-time model depends on the firmware installed in the solver hardware and the number of available eHS licenses in the simulation system. The number of elements in each circuit is subject to the following limitations:

- S = 0..24 switching elements
- U = 1..16 current or voltage inputs
- Y = 1..16 current or voltage measurements
- NSD = 0..60 non-switching devices. The number of non-switching devices includes L and C. "LC" and "RLC" branches each count for two non-switching devices.
- The number of resistors is not limited.

1.6 Intended Audience and Required Skills and Knowledge

The intended user of the eHS solver within OPAL-RT's eFPGAsim Toolbox is an R&D, algorithm, or Test Engineer requiring an easily reconfigurable, very-high-speed electrical circuit solver that does not require knowledge of time-consuming custom firmware development and configuration processes.

1.7 External Tools

1.7.1 Simulink

Simulink is a software package developed by The MathWorks, Inc. that enables modeling, simulation and analysis of dynamic systems. Models are described graphically, following a precise format based on a library of blocks. The eHS solver uses Simulink to define models that will be executed during an RT-LAB simulation. Users are expected to have a clear understanding of Simulink operation, particularly regarding model definition and simulation parameters.

1.7.2 SimPowerSystems and PLECS Simulink toolboxes

The SimPowerSystems and PLECS Blockset toolboxes provide libraries and analysis tools useful for modeling and simulating electrical systems. They are used by the eHS only as circuit description environments. They can be used from within the RT-LAB environment for real-time simulation, but fail to achieve the very high sample rate that can be attained with the eHS solver. It can be useful, though, to use these simulation tools to compare the results to those of the eHS solver.

1.7.3 PSIM

PSIM is a simulation environment engine. It uses a strong algorithm dedicated to electrical circuits (piecewise method, generic models, and a fixed time-step). The fast simulation allows repetitive simulation runs and significantly shortens the design cycle. PSIM's control library provides a comprehensive list of components and function blocks, and makes it possible to build virtually any control scheme quickly and conveniently. It is used by the eHS only as a circuit description environment. The PSIM simulation tools are also useful to compare the results to those of the eHS solver.

1.7.4 NI MULTISIM

NI Multisim is an advanced, industry-standard, best-in-class SPICE simulation environment used by educators, researchers, and engineers worldwide. It is used by the eHS only as a circuit description environment. The NI Multisim simulation tools are also useful to compare the results to those of the eHS solver.

2. REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

Recommended configurations for basic use (only circuit simulation):

- MATLAB 2011b / 2013a / 2015aSP1, 32-bit version
- RT-LAB v11.0.8 or later[5]
- RT-EVENTS v4.0.2 or later[6]
- At least one of this tool:
    - SimPowerSystem "Specialized Technology" Simulink library
    - Powersim PSIM v9.3.4 or v10.0.6 (v10.0.5 is unsupported)
    - Plexim PLECS Simulink Library v3.7.4
    - NI Multisim 13

The RT-XSG toolbox requires the following software to generate programming files for reconfigurable devices and to program the platform:

Recommended configuration for firmware generation:

For Xilinx Zynq and 7 series:

- MATLAB 2014b, 64-bit version
- Xilinx Vivado suite 2015.3 & Xilinx Vivado System Generator for DSP 2015.3
- RT-XSG v3.1.2

For Xilinx Virtex 6 series:

- MATLAB 2011b, 64-bit version
- Xilinx ISE suite v14.7 (System or DSP edition) & Xilinx System Generator for DSP v14.7
- RT-XSG v2.3.5

Complete compatibility charts:

1. MATLAB:

	MATLAB R2011b	MATLAB R2012b	MATLAB R2013a	MATLAB R2013b	MATLAB R2014b	MATLAB R2015aSP1
Basic	X	X	X	X	X	X
Firmware Generation (VC707)	Beta	N/A	N/A	N/A	X	N/A
Firmware Generation (ML605)	X	N/A	N/A	N/A	N/A	N/A
Firmware Generation (OP7XXX)	X	N/A	N/A	N/A	N/A	N/A
Firmware Generation (OP45XX)	Beta	N/A	N/A	N/A	X	N/A
Firmware Generation (OP4200)	Beta	N/A	N/A	N/A	X	N/A

N/A – Unsupported, X – Supported, Beta – Available but not verified

[5] The eHSx32 for OP4200 block is currently only supported as of RT-LAB 11.1.1.

[6] RT-EVENTS is currently unsupported on the OP4200 target.

2. RT-LAB vs Hardware chassis (only circuit simulation):

Chassis	RT-LAB v10.4	RT-LAB v10.5	RT-LAB v10.6	RT-LAB v10.7	RT-LAB v11.0
OP5600 chassis (ML605 board)	N/A	X	X	X	X
OP7000 chassis (OP7161 board)	N/A	X	X	X	X
OP7020 chassis (VC707 board)	N/A	N/A	X	X	X
OP5607 chassis (VC707 board)	N/A	N/A	N/A	X	X
OP4500 chassis (MMPK7 board)	N/A	N/A	X	X	X
OP4510 chassis (TE0741 board)	N/A	N/A	N/A	X	X

N/A – Unsupported, X – Supported

3. RT-XSG vs Hardware chassis (for firmware generation only):

Chassis	RT-XSG v2.1	RT-XSG v2.2.0	RT-XSG v2.2.1	RT-XSG v2.2.2	RT-XSG v2.3.X	RT-XSG v3.0.X
OP5600 chassis (ML605 board)	N/A	N/A	X	X	X	N/A
OP7000 chassis (OP7161 board)	N/A	X	X	X	X	N/A
OP7020 chassis (VC707 board)	N/A	N/A	N/A	X	X	X
OP5607 chassis (VC707 board)	N/A	N/A	N/A	X	X	X
OP4500 chassis (MMPK7 board)	N/A	N/A	N/A	X	X	Beta
OP4510 chassis (TE0741 board)	N/A	N/A	N/A	N/A	X	X

N/A – Unsupported, X – Supported, Beta – Available but not verified

2.2 INSTALLATION NOTES

After installing MATLAB, RT-LAB and RT-EVENTS (plus Xilinx ISE and RT-XSG for firmware generation), install eFPGAsim using the install shield provided. When the installation is done, open a MATLAB instance and type **ver**. The eFPGAsim blockset is correctly installed when it appears in the toolbox list of MATLAB.

```
>> ver
-------------------------------------------------------------------------------
MATLAB Version 7.13.0.564 (R2011b)
MATLAB License Number: 528812
Operating System: Microsoft Windows 7 Version 6.1 (Build 7601: Service Pack 1)
Java VM Version: Java 1.6.0_17-b04 with Sun Microsystems Inc. Java HotSpot(TM) Client VM mixed mode
-------------------------------------------------------------------------------
MATLAB                                                  Version 7.13       (R2011b)
Simulink                                                Version 7.8        (R2011b)
ARTEMIS Blockset                                        Version 7.0.3.874    (R2011b)
Control System Toolbox                                  Version 9.2        (R2011b)
Fixed-Point Toolbox                                     Version 3.4        (R2011b)
MATLAB Coder                                            Version 2.1        (R2011b)
RT-EVENTS Blockset                                      Version 4.0.1.473    (R2011B.x)
RT-LAB                                                  Version v11.1.x.integration.847    (R2011b.x)
Signal Processing Toolbox                               Version 6.16       (R2011b)
SimPowerSystems                                         Version 5.5        (R2011b)
Simscape                                                Version 3.6        (R2011b)
Simulink Coder                                          Version 8.1        (R2011b)
Stateflow                                               Version 7.8        (R2011b)
Xilinx System Generator                                 Version 14.7       production build
eFPGAsim                                                Version v1.5.0.647-4    (R2011b)
>>
```

*Figure 1 – MATLAB command window displaying **ver** command*

3. BUILDING MODELS WITH THE EHS SOLVER

This chapter covers important topics related to the creation of an RT-LAB Simulink model with an eHS solver. It is assumed that the user is already familiar with the RT-LAB and Simulink simulation environments.

3.1 CIRCUIT-UNDER-TEST DESIGN USING THE SIMPOWERSYSTEMS SIMULINK TOOLBOX

3.1.1 CIRCUIT DESIGN USING THE SIMPOWERSYSTEMS LIBRARIES

The circuit must be designed using a special subset of blocks found in the SimPowerSystems libraries. Also, these blocks must be properly named such that they can be easily managed outside the eHS solver.

3.1.2 SUPPORTED BLOCKS

In the SimPowerSystems Elements library, the following blocks are supported (refer to Figure 2):

- Series RLC Branch/Load
- Parallel RLC Branch/Load
- Ground/Neutral
- Breaker
- Pi section lines
- Linear Transformers
- Mutual Inductances
- Inductance Matrix Type Transformers

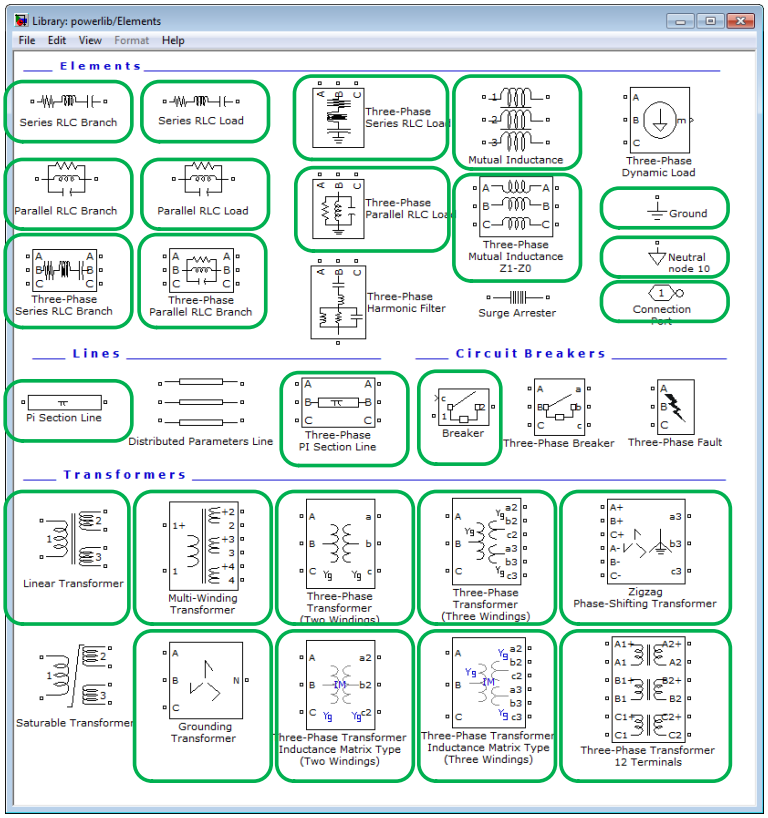


Figure 2 – Elements supported in the SimPowerSystems Elements library

In the SimPowerSystems Measurements block library, the following blocks are supported (refer to Figure 3):

- Current Measurement
- Voltage Measurement
- Three-Phase VI Measurement

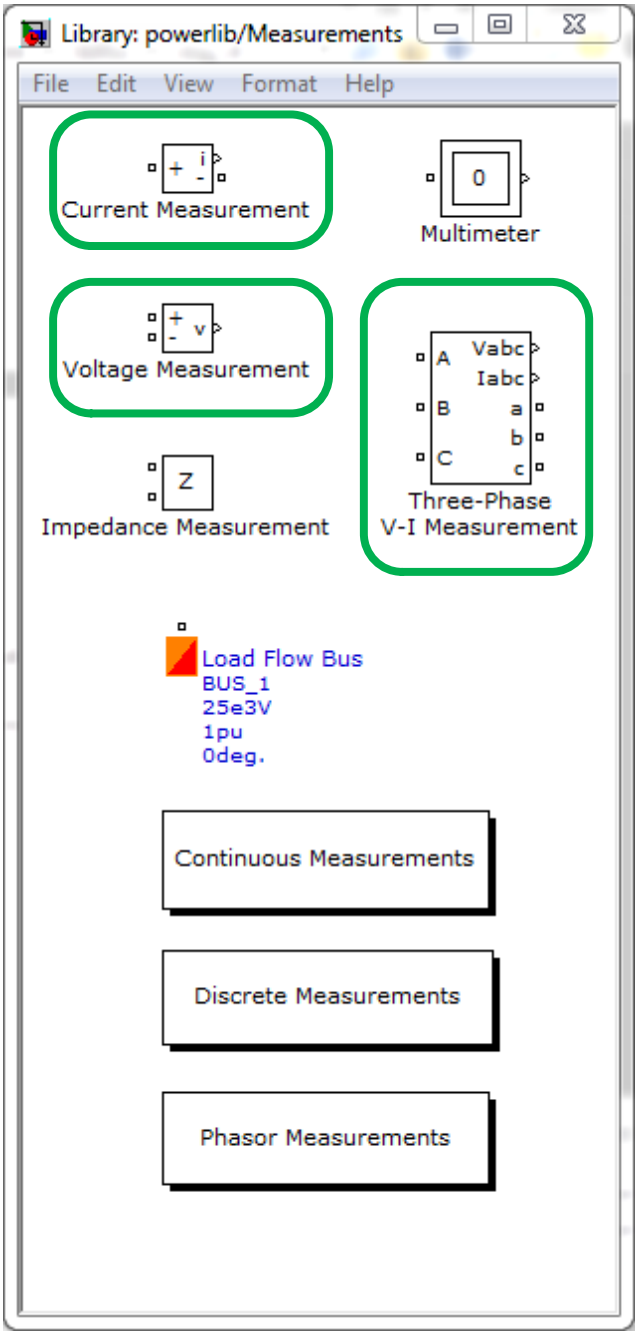


Figure 3 – Elements supported in the SimPowerSystems Measurements block library

In the SimPowerSystems Power Electronics block library, the following blocks are supported (refer to Figure 4):

- Breaker
- Diode
- IGBT
- IGBT/Diode
- Ideal Switch
- Three-Level Bridge
- Thyristor
- Universal Bridge

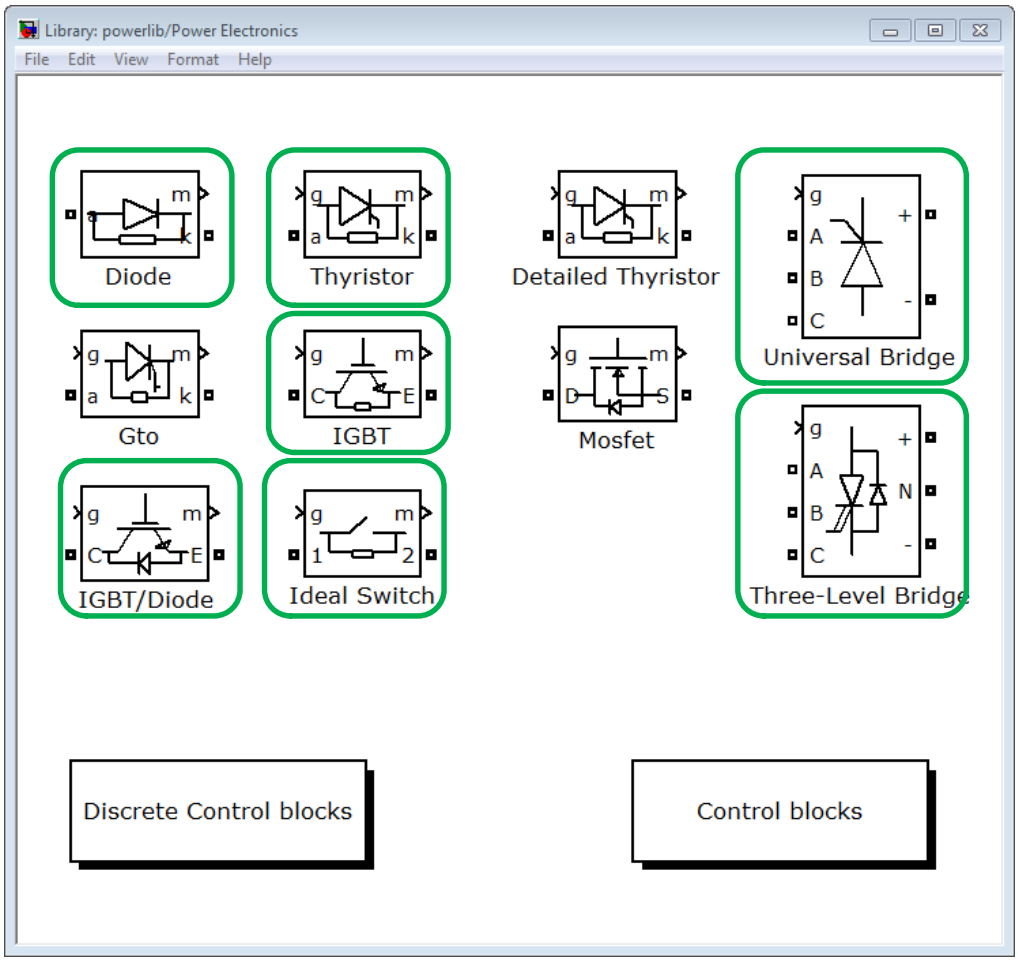


Figure 4 – Elements supported in the SimPowerSystems Power Electronics block library

In the SimPowerSystems Electrical Sources block library, the following elements are supported (refer to Figure 5):

- DC Voltage Source
- AC Voltage Source
- AC Current Source
- Controlled Voltage Source
- Controlled Current Source
- Three-Phase Source

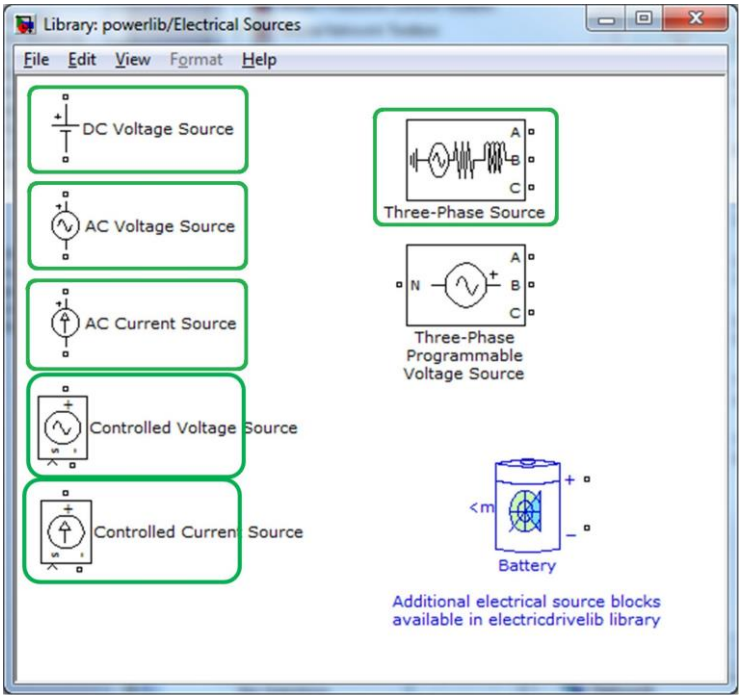


Figure 5 - Elements supported in the SimPowerSystems Electrical Sources block library

3.1.3 COMPONENT NAMING[7]

The algorithm implemented by OPAL-RT (to convert the SimPowerSystems model into the data used by the eHS to compute the currents and voltages in the circuit in real time) requires that the model be analyzed to find the supported elements listed above. For source and switch control signals, as well as measurement outputs, the corresponding elements are listed in alphabetical order and will be accessible in that order within the Simulink/RT-LAB environments. For that reason, it is strongly recommended to rename elements with names that are easily recognizable and whose identification from within RT-LAB will be straightforward. The following naming convention is strongly recommended:

- **Switching devices should be named with the prefix "SW" followed by a 2-digit index**, starting from SW01. Elements categorized as "switching elements" are the following: Diode, IGBT/Diode, Breaker and Ideal Switch. This prefix can be followed by other characters.
- **Sources**, whether controlled current or voltage sources, **should have the prefix "U" followed by a 2-digit index**, starting from U01. This prefix can be followed by other

[7] This naming convention is optional when using the eHSx32 for OP4200 block.

characters. To avoid confusion, the prefix for a three-phase source should contain three indices, starting with phase A (e.g. “U05 U06 U07 Vabc”).

- **Measurements of any kind should have the prefix “Y” followed by a 2-digit index**, starting from Y01. This prefix can be followed by other characters.
- **Inductances, capacitors and resistors** are not accessed directly from RT-LAB and do not have any special naming convention.

Figure 6 below displays an example of a valid SimPowerSystems model for a three-phase inverter:

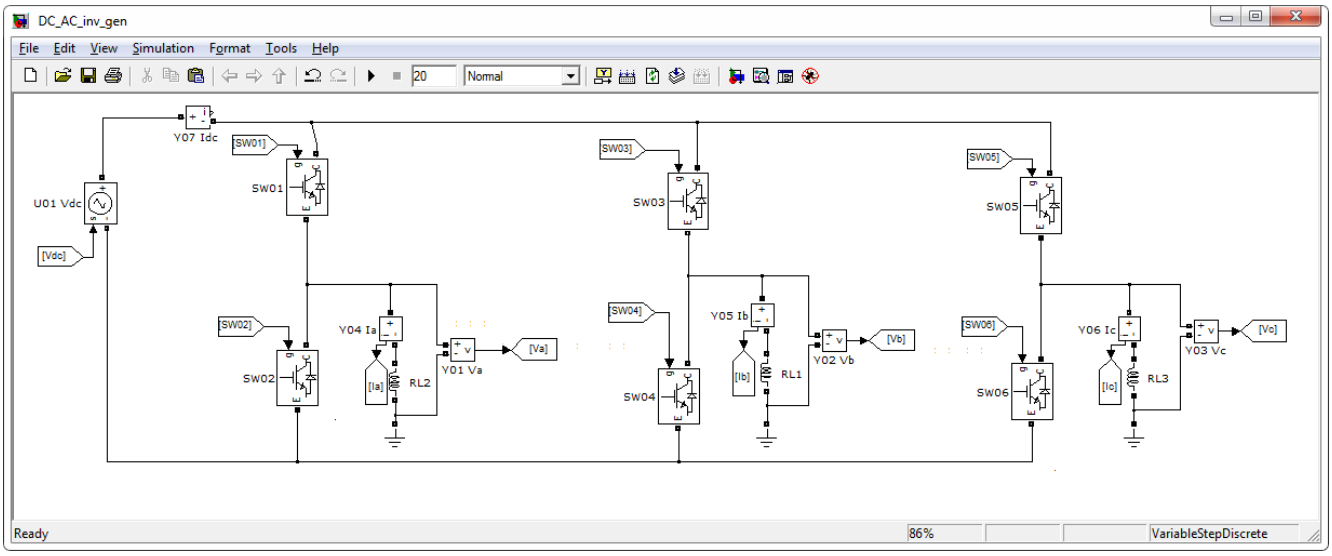


Figure 6 - A three-phase inverter implemented with supported SimPowerSystems elements with block named according to the naming convention described in the chapter.

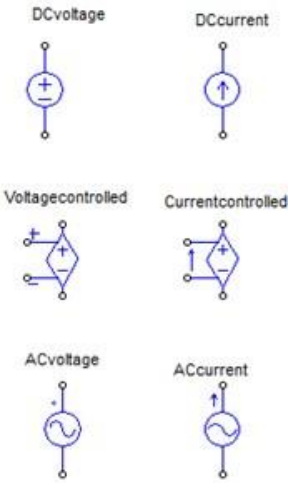
3.2 CIRCUIT-UNDER-TEST DESIGN USING THE PSIM TOOLBOX

3.2.1 CIRCUIT DESIGN USING THE PSIM LIBRARIES

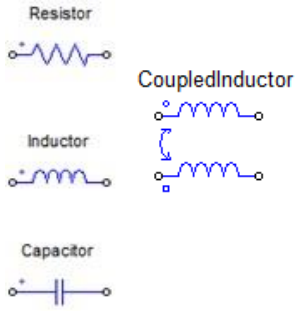
The circuit must be designed using a special subset of blocks found in the PSIM libraries. Also, these blocks must be properly named such that they can be easily managed outside the eHS solver.[7]

3.2.2 SUPPORTED BLOCKS

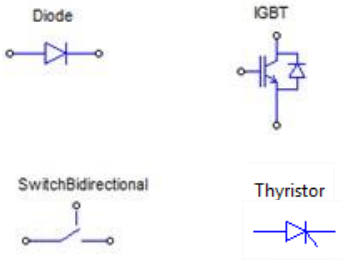
- Sources



- Passive components

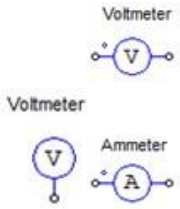


- Switches

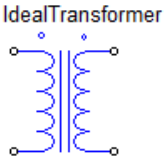


- Ground

- Measurements



- Transformers



3.2.3 COMPONENT NAMING[7]

The algorithm implemented by OPAL-RT (to convert the PSIM model into the data used by the eHS to compute the currents and voltages in the circuit in real time) requires that the model be analyzed to find the supported elements listed above. For source and switch control signals and measurement outputs, the corresponding elements are listed in alphabetical order and will be accessible in that order within the Simulink/RT-LAB environments. For that reason, it is strongly recommended to rename elements with names that are easily recognizable and whose identification from within RT-LAB will be straightforward.

Switching devices should be named with the prefix "SW" followed by a 2-digit index, starting from SW01. Note that the "0" cannot be ignored. Elements categorized as "switching elements" are the following: Diode, IGBT, and Bi-directional switch.

Sources, whether controlled current or voltage sources, should have the prefix "U" followed by a 2-digit index, starting from U01. Note that the "0" cannot be ignored.

Measurements of any kind should have the prefix "Y" followed by a 2-digit index, starting from Y01 and the "0" cannot be ignored.

Inductances, capacitors and resistors are not accessed directly from RT-LAB and do not have any special naming convention.

Below is an example of a valid PSIM model for a Three-Level NPC:

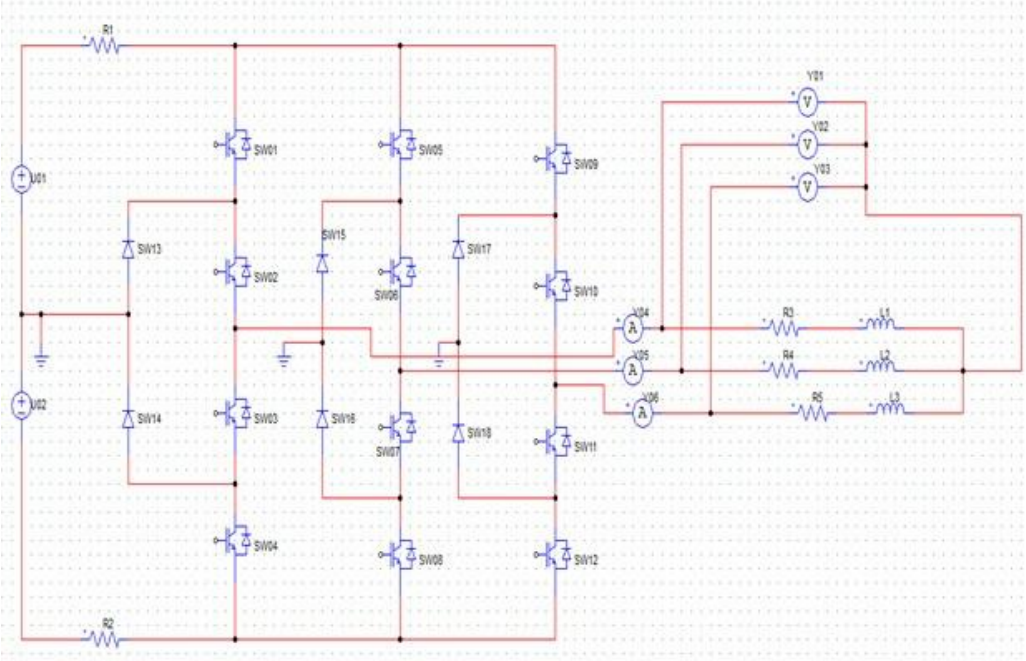


Figure 7 – A three-Level NPC implemented with supported PSIM components

3.3 CIRCUIT-UNDER-TEST DESIGN USING THE PLECS BLOCKSET SIMULINK TOOLBOX

3.3.1 CIRCUIT DESIGN USING THE PLECS LIBRARIES

The circuit must be designed using a special subset of blocks found in PLECS. Also, these blocks must be properly named such that they can be easily managed outside the eHS solver.[7]

3.3.2 SUPPORTED COMPONENTS

Figure 8 shows the supported blocks from PLECS Blockset library.

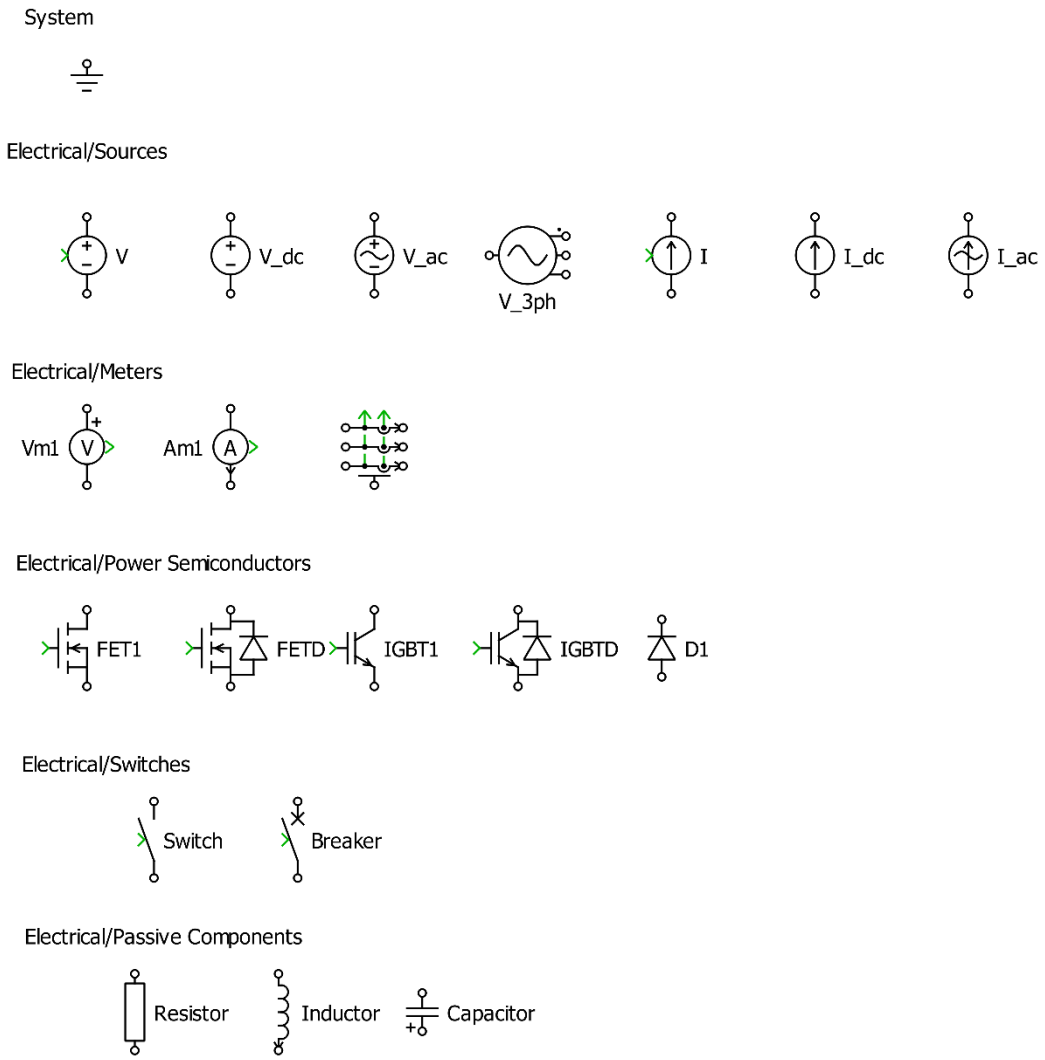
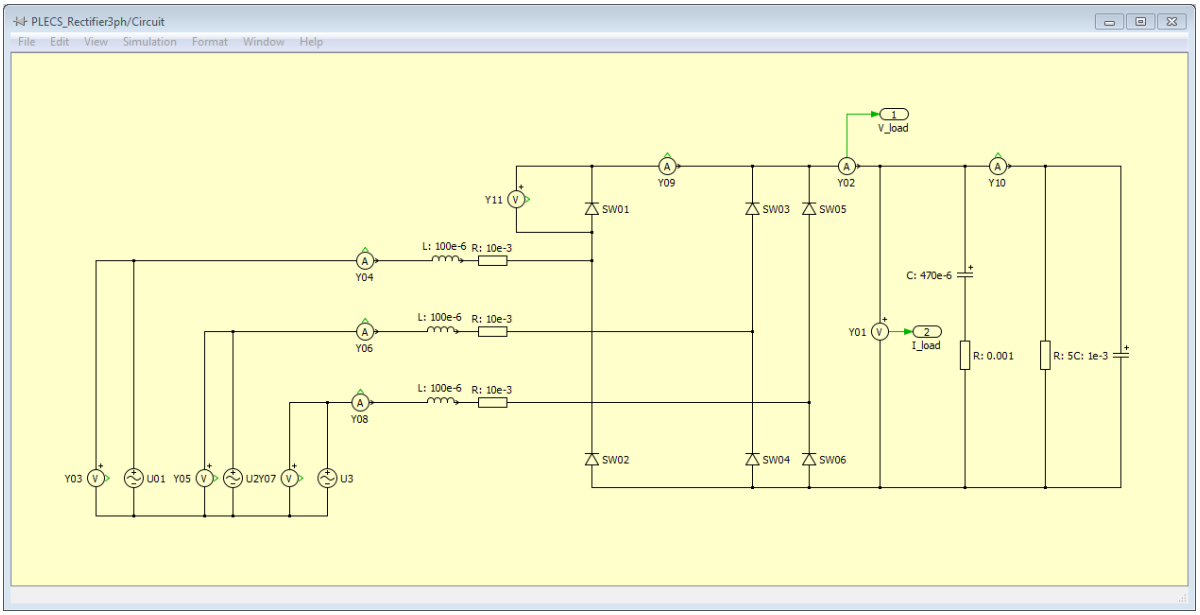


Figure 8 – List of component supported from PLECS Blockset library

3.3.3 EXAMPLE OF CIRCUIT DESIGN USING PLECS

Figure 9 below demonstrates an example of a model design using the PLECS Blockset for use with the eHS.



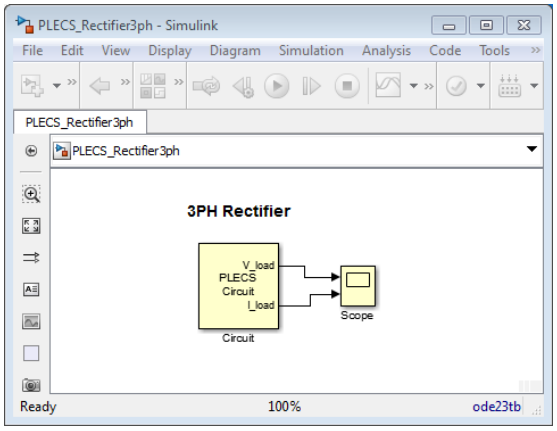


Figure 9 – A three-phase rectifier implemented with supported elements and block named according to the naming rules in PLECS Blockset.

3.4 INSERTING THE CIRCUIT INTO AN RT-LAB MODEL

3.4.1 EHS BLOCKS

The circuit described by a SimPowerSystems model, PLECS Blockset model, or a PSIM design (*.psimsch) is imported using an eHS block. The different eHS blocks can be found in the *eFPGAsim/eHS and Converter Models* Simulink Library section.

The eHS blocks implement the driver that manages all the communication within the eHS firmware including solver initialization and the transmission in real time of the circuit control signals (current and voltage source control signals, switching information of the transistors, breakers, and other switches).

The block must be placed in an RT-LAB compatible model that will be compiled and executed by the RT-LAB software. Templates are provided in the eFPGAsim installation (path: *C:\OPAL-RT\EFPGASIM\<version>\Examples\eHS_withIOs\<chassis>\Simulink\rtlab*). If your chassis is not listed, please contact OPAL-RT Support for more information about the supported hardware configuration.

3.4.1.1 THE EHSX32 GEN3 BLOCK FOR THE OP4200

Figure 10 below displays the eHSx32 for OP4200 block.

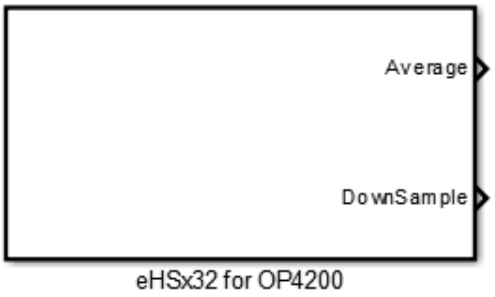


Figure 10 - eHSx32 Gen3 block for OP4200

The eHSx32 for OP4200 block implements a different user interface for configuration than the eHS Gen3 block to improve usability but contains many of the same features.[8] Since a different user interface was developed for the OP4200, the naming convention described in the section on Building Models with the eHS Solver is not critical. The Average and Downsample outputs of the block are returned as vectors containing the measurements of the Circuit Model (see below for information on the Outputs). The order of the elements is defined in the Outputs tab.

The figures below describe the tabs of the eHSx32 for OP4200 block:

- The Circuit tab of the block allows the user to configure the Circuit Model file being simulated and the Time Step information of the solver. It also contains a table of information about the Circuit Model, including the number of Sources, number of Measurements, number of Switches, and number of States. During selection, the Circuit Model is parsed automatically, allowing the other tabs of the block to be populated with information relating to the Circuit Model.

[8] The Scenarios feature, for example, is unsupported. For a complete description of the specifications of the eHSx32 gen3 core, see Table 1.

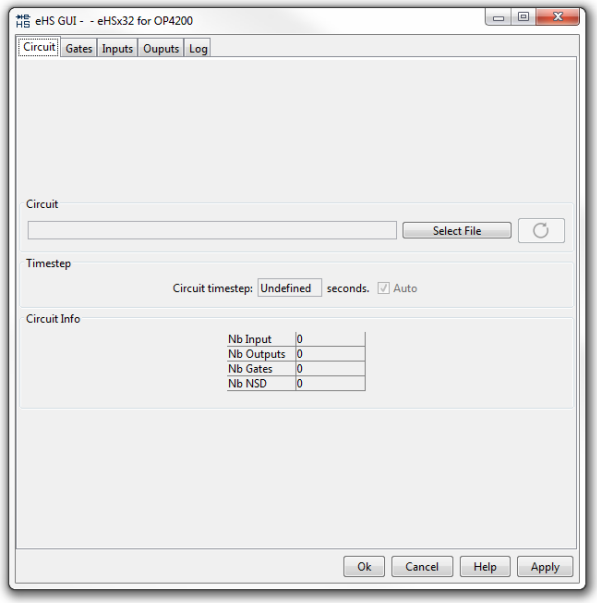


Figure 11 - Circuit Tab of the eHSx32 for OP4200 block

- The Gates tab allows the user to configure the Switch Source Type & Channel for each switch (either from **Digital In** channel or from the **CPU**[9]), the polarity of each switch (**High** or **Low**), and the Gs value of each switch.[10] By selecting **CPU**, the block regeneration will expose this gate as an input port to the block.

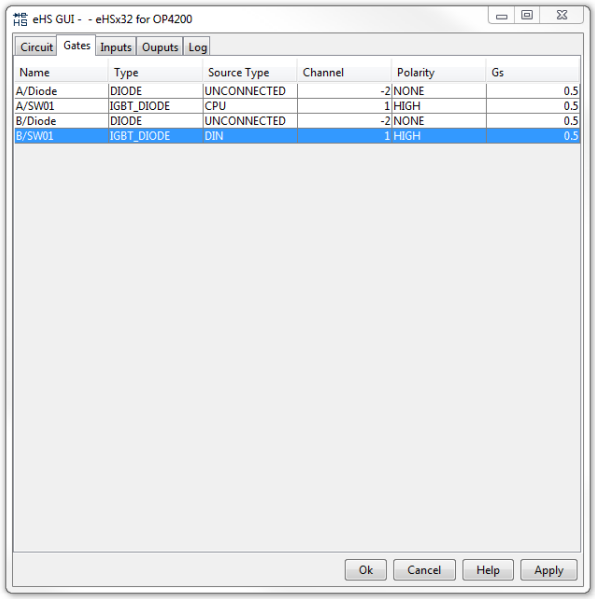


Figure 12 - Gates tab of the eHSx32 for OP4200 block

[9] In order to use the firmware's internal PWM generators as the source of a Gate in the eHS, the PWM Generators should be configured on a Digital Out channel in RT-LAB using the OPAL-RT Board driver, a Digital In Channel should be configured in the eHS block, and a loopback will need to be made physically between the Digital Out and Digital In pins. For more details, please refer to the RT-LAB User Manual and the eFPGAsim Quickstart Guide.

[10] Diodes are considered disconnected as their Source Type, Channel, and Polarity are not configurable. Only their Gs values can be edited by the user. Note that when using the two-level and three-level LCA, the Gs values need to be the same for each switch in the component.

- The Inputs tab allows the user to configure the Current and Voltage Sources used in the circuit model including the Signal Source (**CPU** or **Constant** value), as well as the parameters (used to configure the value of the Constant) of each input. By selecting **CPU**, the block regeneration will expose this input as an input port to the block.

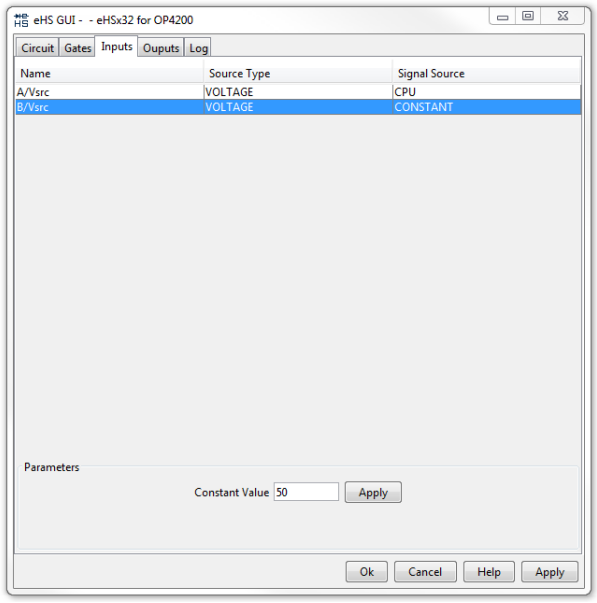


Figure 13 – The Inputs tab of the eHSx32 for OP4200

- The Outputs tab allows the user to view the Measurement points configured within the Circuit Model.

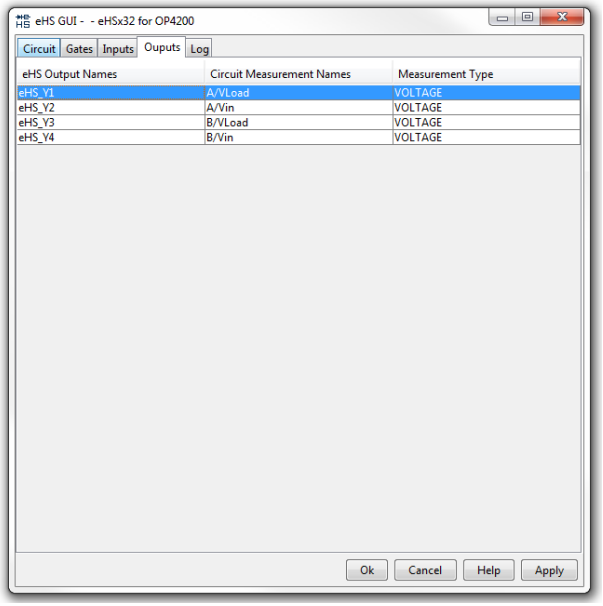


Figure 14 - The Outputs tab of the eHSx32 for OP4200

- The Log tab allows the user to view debugging information on the parsing of their circuit model and the stream generation.

Please refer to the Help for the block for more details on describing the inputs and outputs of this block.

3.4.1.2 THE EHS GEN3 BLOCK

Figure 15 below displays the eHS Gen3 block. Currently, the block supports the x32, x64, and x128 form factors.

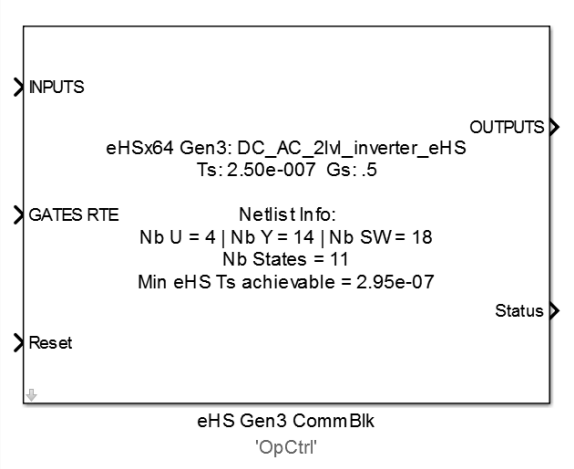


Figure 15 – eHS Gen3 block

- The INPUTS port of the eHS Gen3 block accepts a vector containing Voltage & Current Source values as defined by the Circuit Model. If the naming convention described in Figure 6 was used in the circuit model, the index of each element in the input vector will correspond to the source index of the circuit model.[11]
- The GATES RTE/GATES STATIC port of the eHS Gen3 block accepts a vector containing Switch control values as defined by the Circuit Model. The data type of each element can either be of type RTE (for the GATES RTE port) or Double floating-point representation (for the GATES STATIC port). If the naming convention described in Figure 6 was used in the circuit model, the index of each element in the input vector will correspond to the source index of the Circuit Model.[11]
- The OUTPUTS port of the eHS Gen3 block returns a vector of the Voltage and Current Measurement values as defined by the Circuit Model. If the naming convention described in Figure 6 was used in the circuit model, the index of each element in the output vector will correspond to the measurement block index in the circuit model.[11] [12]

For a detailed overview of this block, please refer to the Help for the block.

3.4.2 SIGNAL TYPES

The eHS solver input and output signal types, as seen from RT-LAB, are of Double Floating-Point data representation. The signals are converted within the block to single precision or extended single precision. The conversion to single precision enables the solver to run faster, but can cause discrepancies between the results obtained by the eHS and those obtained by the SimPowerSystems simulation for circuits with very long time constants relative to the sample time of either solver.

[11] The naming is done by alphabetical order. Therefore, if the user is using their switches inside subsystems, this will have an effect on the order. To observe the order of sources, switches, and measurements, please refer to the Infos tab of the eHS Gen3 block or the appropriate tab on the eHSx32 for OP4200 block.

[12] It is to be noted that while the Dual eHS block has zero hold sampled output, the outputs of eHS gen3 block are averaged over the previous CPU time step (i.e. sum of all the FPGA core output samples during the CPU time step divided by the number of samples).

3.4.3 OFFLINE SIMULATION

The eHS block does not enable offline simulation. For offline simulation, the appropriate block in the eFPGAsim eHS and Converter Models Simulink Library must be used. This block enables the developer to connect the block exactly as it is connected inside the FPGA-based board firmware (e.g. to a plant model) for more accurate results. One offline block should be added for each solver implemented. Connections between them and any other component present in the firmware model should be done manually by the developer. This also enables the developer to simulate each part of the circuit with the appropriate sample time, especially the faster-running eHS solver.

3.4.4 UNDERSTANDING SAMPLE TIMES

The eHS solver sample time is accessed through a parameter of the eHS block. By default, this sample time is set to the minimum sample time of the solver according to the circuit-under-test. This value depends upon the complexity of the circuit, and ranges typically from 160 nanoseconds to 4 microseconds. A higher value can also be explicitly given, using the appropriate option in the Dual eHS block parameter panel.

The inputs and outputs of the eHS solver are sampled at the sample time indicated in the eHS parameter panel. Nevertheless, inputs coming from, and outputs received by the RT-LAB model will be sampled at the RT-LAB sample time, which is typically longer (see Figure 16 below).

Gated signals sent using the RT-EVENTS signal type “RTE Boolean” have a time resolution higher than the RT-LAB model sample time, and are thus sampled at the eHS sample time.[6]

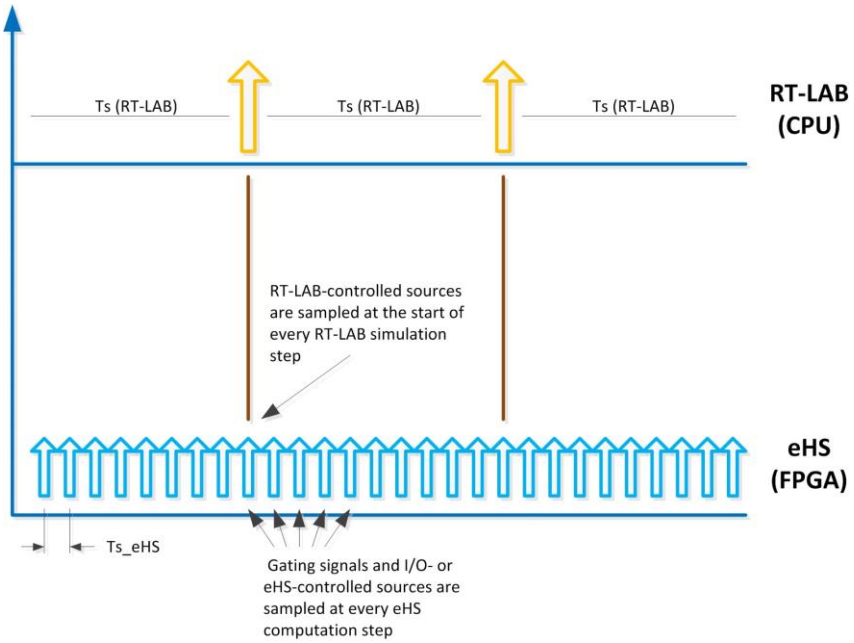


Figure 16 – Sample times: eHS runs at a faster sample time, but inputs coming from RT-LAB arrive at a slower rate.

Because of inherent communication procedures between the FPGA-based hardware and RT-LAB, a latency of two RT-LAB time steps is observed between the time when commands are sent to the eHS and when its response in the circuit can be read again in RT-LAB, as shown in 0below:

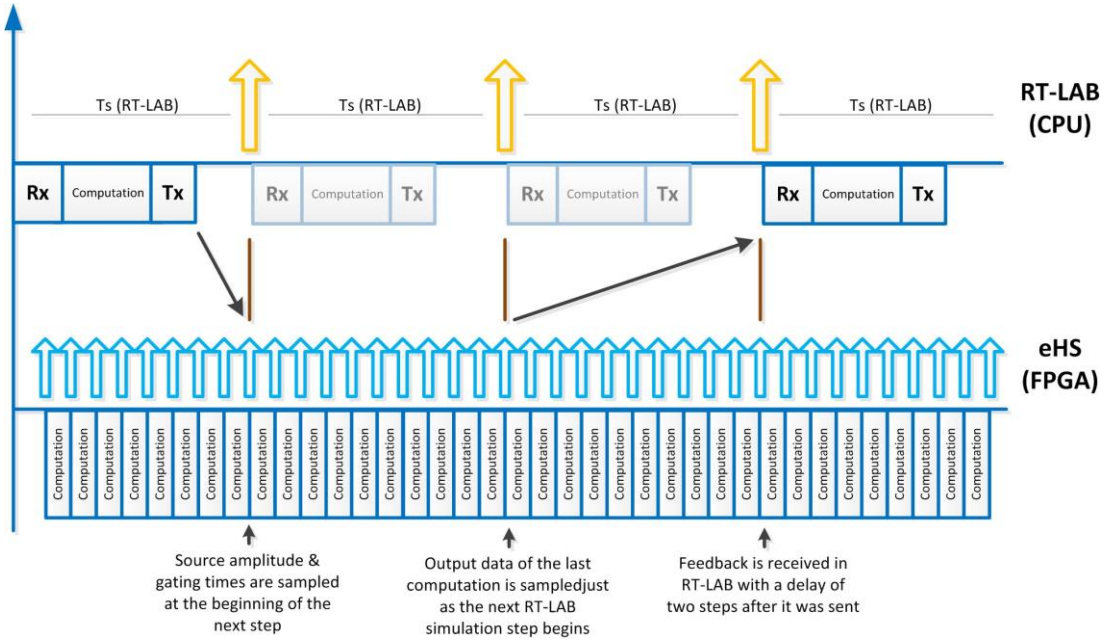


Figure 17 – Feedback data is received within RT-LAB two simulation steps after commands were sent to the eHS solver

3.5 INTERCONNECTING MULTIPLE EHS SOLVERS

As previously explained, gated signals have a time resolution higher than the RT-LAB simulation step size. These signals are typically pulse-width modulated signals (PWM), and are generated by the FPGA firmware with a time resolution in the order of 10 nanoseconds. Hence, these signals can be sampled at the eHS computation rate. On the other hand, input signals generated by RT-LAB will, by default, be sampled with a sampling period equal to the RT-LAB simulation step size, which is in the order of 10-50 microseconds. A better resolution can be achieved either by generating the signals directly on the FPGA, by taking inputs from the external world via an analog input interface, or by interconnecting multiple eHS solvers.

By default, eHS sources are accessed through the eHS block and are sent from the CPU of the simulator.

As illustrated in Figure 18 below, the source control signals of either eHS can come from RT-LAB, from the output of the other eHS solver, or from any other FPGA-based solver. The selection is made for each individual signal in the “Input Settings” tab by routing the control of a source by another eHS output.

Note that a diagram of the firmware is required to be sure of the solvers interconnection settings.

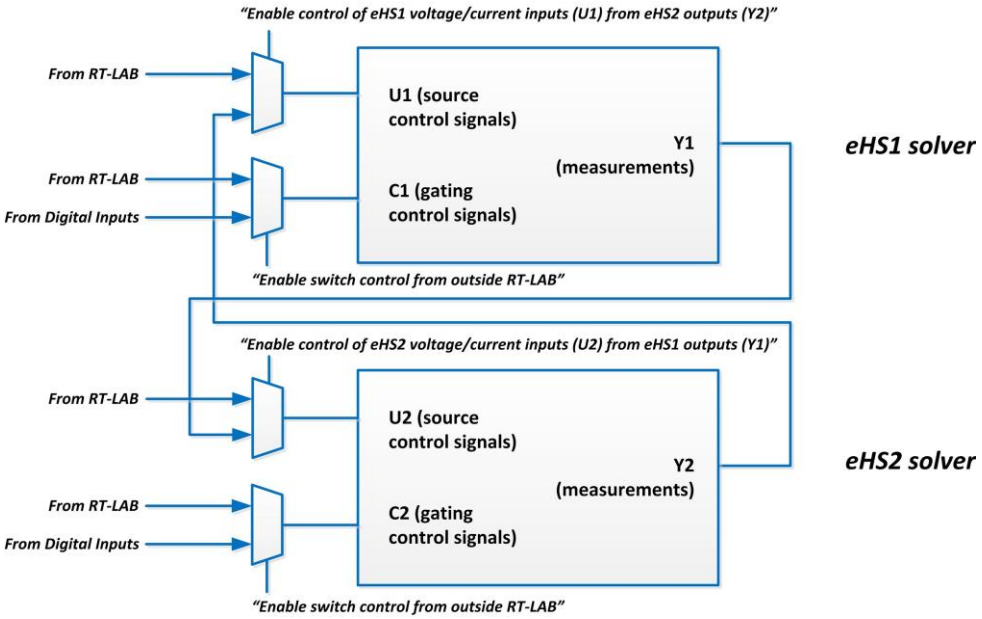


Figure 18 – Interconnection pattern between the two eHS solvers with a custom firmware.

Note: FPGA-based sinusoidal source or gating signal generators as well as other solver interconnection patterns such as machines models can be implemented with the help of the RT-XSG toolbox or with the help of OPAL-RT development team.

3.6 USING THE SCENARIO FEATURE

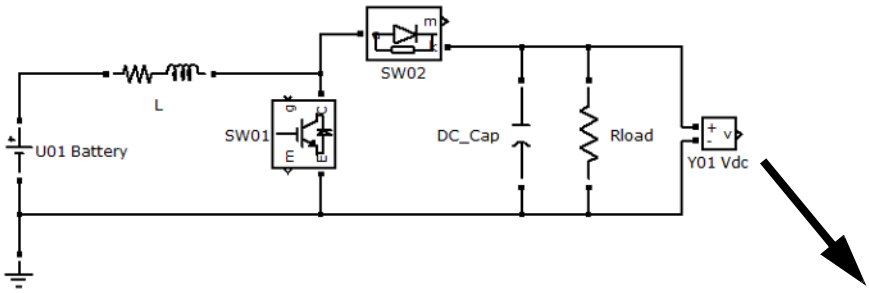
3.6.1 WHAT IS A SCENARIO?

A scenario is a version of the netlist that has its own parameter set (i.e. Capacitors/Inductances/Resistances).

The scenarios feature makes it possible to have multiple versions of the netlist stored in the FPGA solver core. Thus, during runtime, the user is able to switch from one scenario to another in order to modify the model behavior. For example, this could allow the user to apply short or open circuit faults.

The scenarios are managed inside an XLS file. For each scenario, a line is declared, and for each component involved in a scenario, a column is added in the XLS file. For a given scenario, it is possible to modify as many components values as required by the scenario.

3.6.2 WRITING THE XLS SCENARIO FILE



1. Build the netlist

2. Build the Scenario Table

	A	B	C	D	E	F	G	H	I
1		DC_Cap	Rload	L.R	L.L	snubber: SW02	snubber: SW01	SW02_Ron	SW01_Ron
2	Default	0.001	10	0.02	0.001	100000	100000	0.001	0.001
3	Scenario1				0.002				
4	Scenario2	0.002							
5	Scenario3	0.002			0.002				
6	Scenario4		1						
7	Scenario5							1.00E+05	
8	Scenario6								
9	Scenario7								
10	Scenario8								
11	Scenario9								
12	Scenario10								
13	Scenario11								

```
Scenarios found:
#01 #02 #03 #04 #05 #06

Scenario1:'L' L value updated to 0.002
Configuration of scenario1 generated successfully.

Scenario2:'DC_Cap' C value updated to 0.002
Configuration of scenario2 generated successfully.

Scenario3:'DC_Cap' C value updated to 0.002
Scenario3:'L' L value updated to 0.002
Configuration of scenario3 generated successfully.

Scenario4:'Rload' R value updated to 1
Configuration of scenario4 generated successfully.

Scenario5:'SW02_Ron' R value updated to 100000
Configuration of scenario5 generated successfully.
```

3. Generate the eHS matrix

Figure 19 - Scenario Creation Workflow

3.6.2.1 GETTING STARTED

The easiest way to write the first XLS file is to generate a template by selecting the **Create a XLS template for the current netlist** option from the Scenario Management tab. Users can also write the XLS file themselves. The first line, from cell B1, is reserved for RLC components declaration. The first column, from cell A2, is reserved for the scenarios declaration.

3.6.2.2 FILLING THE COMPONENT LINE

The first line of the table, starting at cell B1, is reserved for the component declarations. The component name is the netlist name (from the top-level of the netlist, e.g. **subsystem/componentname**). If the component is inside a branch, its name will be the branch name with the suffix ".R", ".L" or ".C", depending on the type of the targeted component.

3.6.2.3 FILLING THE SCENARIO COLUMN

The first column of the table, from cell A2, is the scenario declaration. Scenario labels must respect the following naming convention: **Scenario** followed by the scenario number (e.g. the correct scenario 21 will be labelled **Scenario21**).

3.6.2.4 DEFAULT SCENARIO LINE

The **Default** label is reserved to display the default values of each component. However, these are not applied to the netlist as default values, but are only there for information. Instead, the netlist file components values are applied by default.

3.6.2.5 REMOVING LINES/COLUMNS

Removing lines (scenarios) and columns (components) from the table is allowed. As a result, the removed scenarios and components will be kept at default values.

3.6.2.6 FILLING THE TABLE

For each scenario, the component parameters to be modified need to be defined. These scenario component values are to be entered in the table. Leaving a cell empty will keep the component's value unchanged by the scenario, thus maintaining its default value.

3.6.2.7 NUMBER OF MODIFICATIONS ALLOWED BY SCENARIO

For each scenario, all declared components' values can be modified. There is no limitation on the number of components that can be edited by each scenario.

3.6.2.8 NON-EXISTING COMPONENT

If a component is declared in the XLS file, but does not exist inside the netlist (e.g. if there is the wrong label or a component was removed), this column will be ignored and this will not have any effect on the scenario generation.

3.6.2.9 NON-EXISTING SCENARIO DECLARATION

If a scenario is impossible (wrong label or scenario number greater than the maximum number of feasible scenarios), an error will be thrown.

	A	B	C	D	E	F	G	H	I
1		DC_Cap	Rload	L.R	L.L	snubber: SW02	snubber: SW01	SW02_Ron	SW01_Ron
2	Default	0.001	10	0.02	0.001	100000	100000	0.001	0.001
3	Scenario1				0.002				
4	Scenario2	0.002							
5	Scenario3	0.002			0.002				
6	Scenario4		1						
7	Scenario5							1.00E+05	
8	Scenario6								
9	Scenario7								
10	Scenario8								
11	Scenario9								
12	Scenario10								
13	Scenario11								

Figure 20 – Table Example

3.6.2.10 VERIFYING THAT SCENARIOS HAVE BEEN PROPERLY GENERATED

While generating the eHSx64 configuration, a log is generated in the MATLAB command prompt to advise the user of the changes made to the netlist depending on the scenario number.

```
Scenarios found:
#01 #02 #03 #04 #05 #06

Scenario1:'L' L value updated to 0.002
Configuration of scenario1 generated successfully.

Scenario2:'DC_Cap' C value updated to 0.002
Configuration of scenario2 generated successfully.

Scenario3:'DC_Cap' C value updated to 0.002
Scenario3:'L' L value updated to 0.002
Configuration of scenario3 generated successfully.

Scenario4:'Rload' R value updated to 1
Configuration of scenario4 generated successfully.

Scenario5:'SW02_Ron' R value updated to 100000
Configuration of scenario5 generated successfully.
```

3.7 THE LOSS COMPENSATION ALGORITHM (LCA) FEATURE (EHS GEN3 ONLY)

3.7.1 WHAT IS THE LCA?

The Loss Compensation Algorithm (LCA) feature removes the power losses that might occur when an IGBT/Diode switch is used (as a result of the Pejovic method). This technique works only for specific converter topologies.

Currently, only eHS Gen3 supports this feature.

3.7.2 RECOGNIZING THE GEN3 BLOCK

First, the Gen3 block must be used to access this feature.

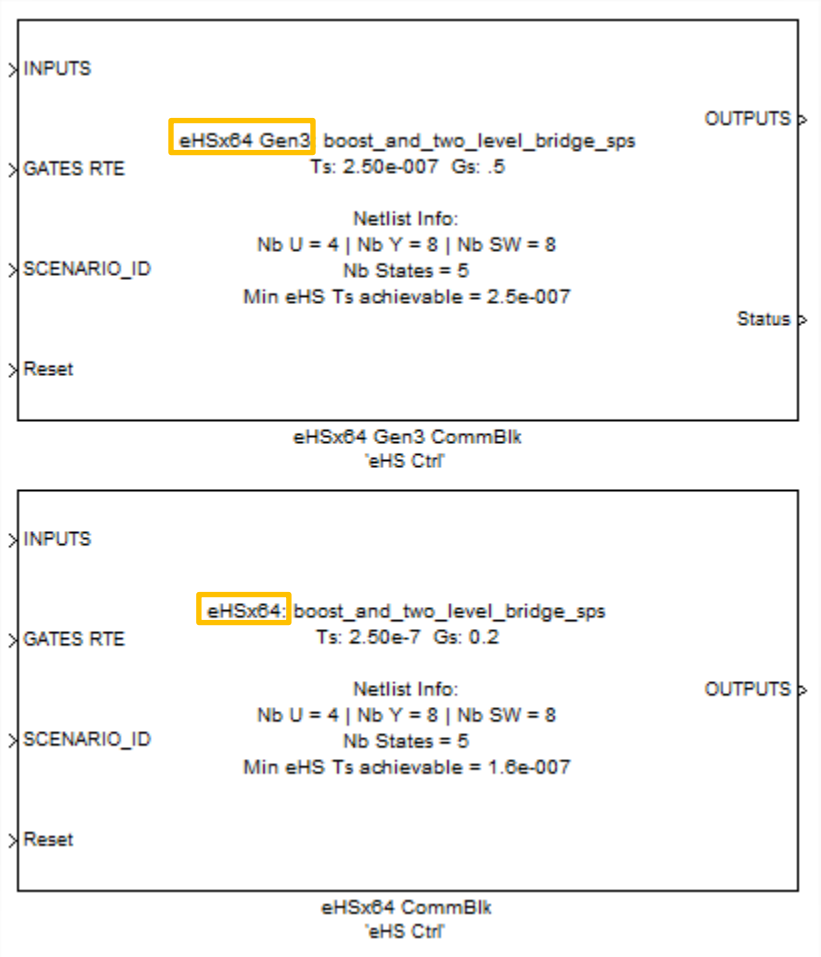


Figure 21 – eHS Gen3 (top) vs eHS Gen2 (bottom)

As shown in Figure 21 above, the Gen2 and Gen3 eHS blocks are very similar. The main difference is the support for the LCA feature that is only accessible with the eHSx64 Gen3 block.

3.7.3 SUPPORTED TOPOLOGIES

The following two topologies support the LCA feature:

- The two-level arm converter

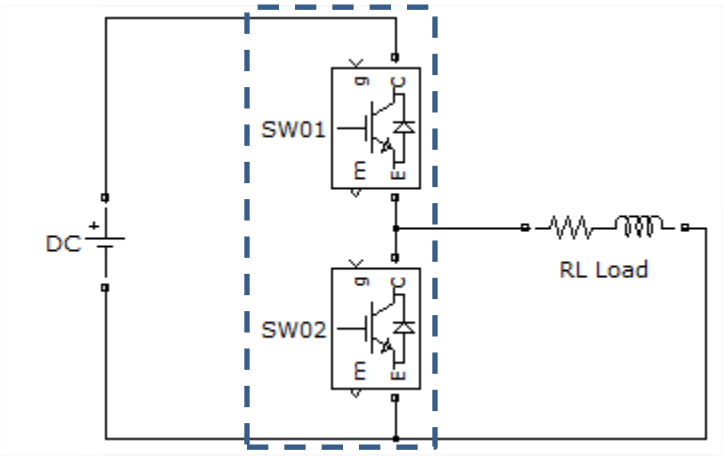


Figure 22 – A two-level arm converter example

- The three-level NPC arm converter

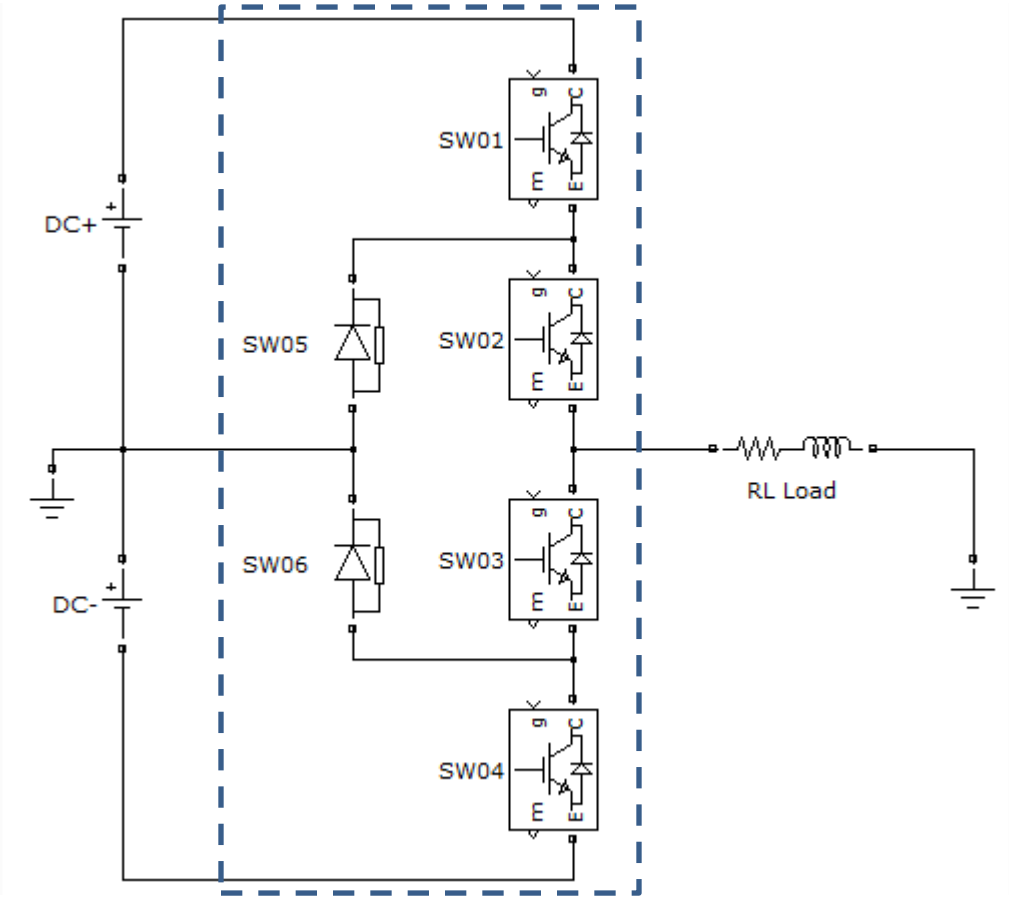


Figure 23 – A three-level NPC arm converter example

3.7.4 How to Use the LCA Feature (SimPowerSystems workflow)

3.7.4.1 Using LCA With a Two-Level Arm Converter

To use the LCA in the SimPowerSystems environment the two-level arm converter should be replaced by the universal bridge component in the eHS circuit model.

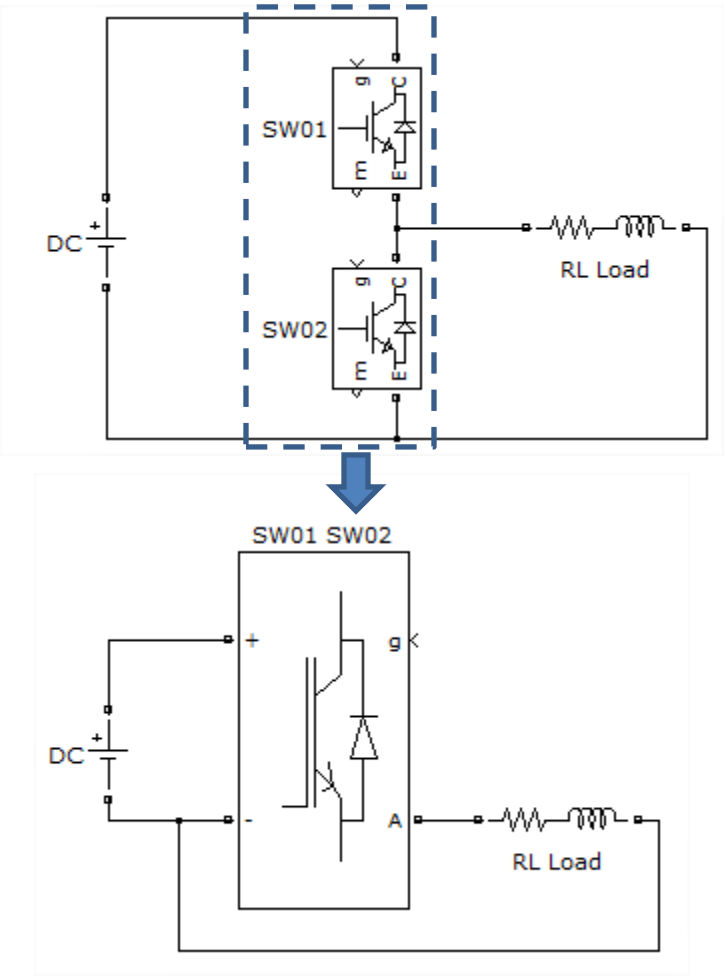


Figure 24 – Using the the LCA feature on a two-level arm converter

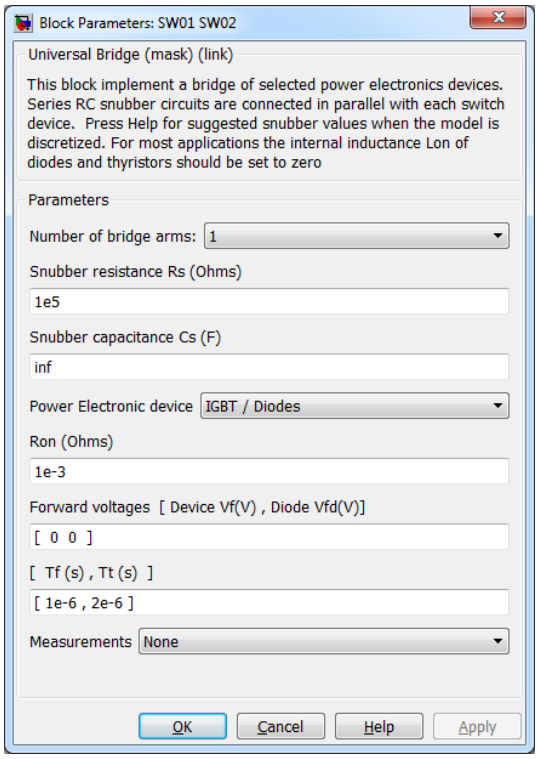


Figure 25 – Universal bridge options

Note that the forward voltages and Tf / Tt parameters will be discarded during the parameter extraction. Only IGBT/Diodes and Diodes are supported in the **Power Electronic device** dropdown list.

Once the circuit model has been updated with the universal bridge component, the user should rebuild the CPU-model in RT-LAB. During the eHS circuit parsing process, the universal bridge will be detected as an LCA element and the loss-compensation algorithm will be used instead of the classic switching logic.

3.7.4.2 USING THE LCA FEATURE ON A THREE-LEVEL NPC ARM CONVERTER

To use the LCA feature in the SimPowerSystems environment, the three-level NPC arm converter should be replaced by the Three-Level Bridge component in the eHS circuit model.

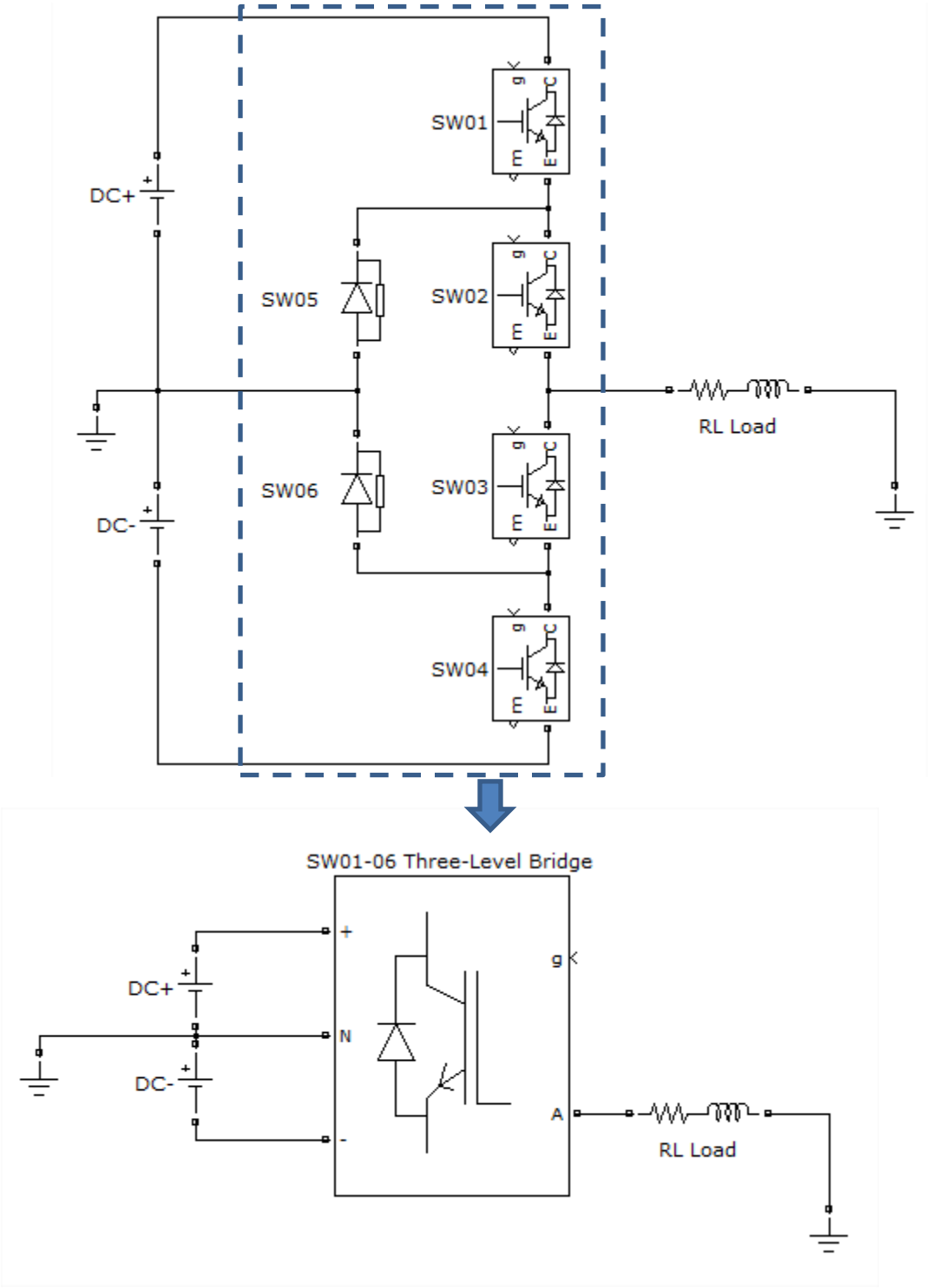


Figure 26 – Use LCA on a three-level NPC arm converter

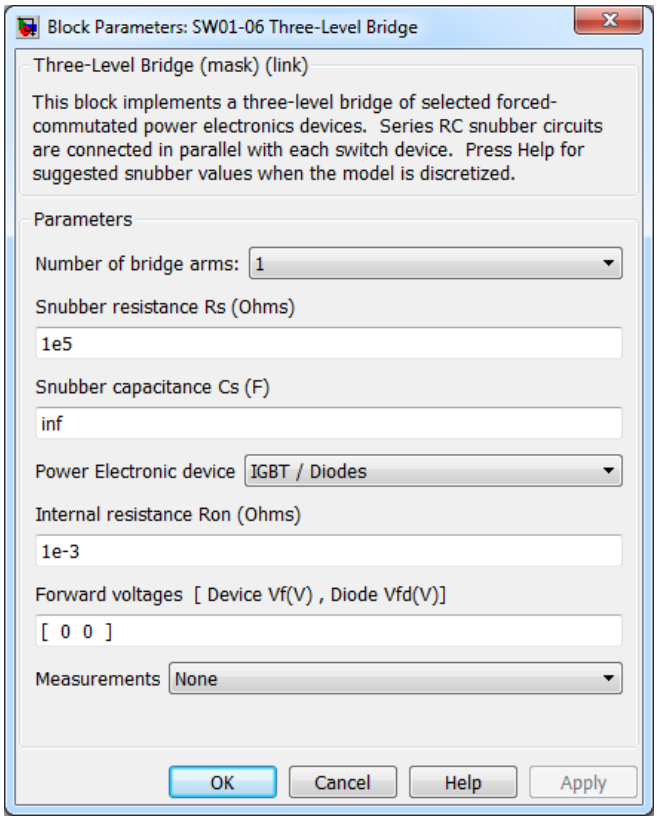


Figure 27 – Three-Level Bridge options

Note that the forward voltage parameters will be discarded during the parameter extraction. Only the **IGBT/Diodes** option is supported in the **Power Electronic device** dropdown list.

Once the circuit model is updated with the Three-Level Bridge component, the user can rebuild the CPU-model in RT-LAB. During the eHS circuit parsing process, the Three-Level Bridge component will be detected as an LCA element and the loss-compensation algorithm will be used instead of the classic switching logic.

3.8 EXAMPLE MODELS AND TEMPLATES

All eFPGAsim example models and template can be found in MATLAB>>Help>>eFPGAsim Blockset Demos or in the installation folder at the following path:

<eFPGAsim Installation Directory>/Examples ,

where the eFPGAsim installation directory is located at *C:/OPAL-RT/eFPGAsim* by default.

Each example contains its own Help documentation located in the CPU model. Please refer to these models for getting started with the eHS.

CONTACT

OPAL-RT Corporate Headquarters

1751 Richardson, Suite 2525

Montréal, Québec, Canada

H3K 1G6

Tel.: 514-935-2323

Toll free (US and Canada): 1-877-935-2323

Fax: 514-935-4994

Technical Services

For additional information you may contact the Customer Support team at OPAL-RT at the following coordinates:

www.opal-rt.com/support

Note:

While every effort has been made to ensure accuracy in this publication, no responsibility can be accepted for errors or omissions. Data may change, as well as legislation, and you are strongly advised to obtain copies of the most recently issued regulations, standards, and guidelines.

This publication is not intended to form the basis of a contract.

eFPGAsim v1.5 : eHS User Guide

01/03/2017