Fall 2021

Real Time Simulation and Hardware in the Loop Methods for Power Electronics Power Distribution Systems

Michele Difronzo

REAL TIME SIMULATION AND HARDWARE IN THE LOOP METHODS FOR POWER ELECTRONICS POWER DISTRIBUTION SYSTEMS

by

Michele Difronzo

Bachelor in Mechanical Engineering
Politecnico di Bari, 2007

Master in Mechanical Engineering
Politecnico di Bari, 2011

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2021

Accepted by:

Andrea Benigni, Major Professor

Herbert L. Ginn III, Committee Member

Roger A. Dougal, Committee Member

Jason D. Bakos, Committee Member

Tracey L. Weldon, Vice Provost and Dean of the Graduate School

DEDICATION

To my lovely wife, my mom, my father, and my aunt Teresa who have always supported me during the most important moments of my life.

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Dr. Andrea Benigni, and my co-advisor Dr. Herbert L. Ginn for the trust and support shown to me during these years especially in the most difficult moments. Likewise, I would like to thank Dr. Roger Dougal and Dr. Jason Bakos for their important contribution as committee members for this dissertation, as professors during my research work and in general for their human and professional support during the last years.

I also would like to acknowledge Dr. Enrico Santi -for supporting me during the classwork period which due to my mechanical engineering background has proved to be very challenging-, the Graduate Coordinator Jenny Balestrero and the Program Coordinator Hope Johnson.

I would like to acknowledge Matthew Milton and Matthew Davidson, whose help has been decisive during this journey, Aaron De La O, Andrew Wunderlich, Biswas Multan, Dhiman Chouwdhury, Haydeer Habbood and Yan Chen, who have always proved to be great friends and colleagues.

I am grateful to Prof. Giuseppe Pascazio, Prof. Michele Napolitano, Prof. Umberto Galietti and Prof. GeCheng Zha to be great mentors, to point me in the right direction and support my applications to the Graduate School.

Finally, I would like to thank my wife, my mom, my father (R.I.P.) and my aunt for being always there supporting and helping me during this journey.

ABSTRACT

System level testing of Power Electronics Power Distribution Systems (PEPDS) can be challenging when fine temporal resolution is required (time step below 100-200ns). In the recent years, our research group has proposed various methods to simulate in real-time PEPDS using FPGAs and time step as small as 50ns. While the proposed methods allow achieving the desired temporal resolution, they are extremely demanding in terms of resources usage and the size of the PEPDS that can be simulated on a single FPGA is strongly limited.

In this dissertation -work that takes as an example application the US Navy electric Ship Zonal System (SZS)- a platform based on a commercial CPU based simulator and on a custom multi-FPGA simulator is presented. The multi-FPGA simulator enables system level PEPDS analysis while maintaining a very small-time step (70ns).

Using a CPU commercial platform and multi rate execution, the power electronics part of the system is simulated together with the slow electro-mechanical portion of the PEPDS maintaining a unified vision.

To achieve such a small simulation time step, the LBLMC method is applied and an innovative parallel bus interface architecture for a three FPGAs layout is introduced.

The PEPDS model is decomposed for multi-FPGA executions using the nodal decomposition method. Two converters models, MMC and DAB, have been developed and included in the Open Real-Time Simulation (ORTiS) framework.

To allow multi-rate execution a dedicated software and hardware interface has been developed so to interface the custom FPGA based simulator -operating with a 70ns time step- with the commercial CPU based simulator -operating at 25μs.

Furthermore, to increase the flexibility and scalability of the proposed simulation platforms, a co-simulation interface based on the Aurora protocol and realizing communication between a multi-CPU and a multi-FPGA based platforms is introduced.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CI..... Current Injection

CDC .....Clock Domain Crossing

CHIL .....Control Hardware In the Loop

CTRL ..... Control

DAB ..... Dual Active Bridge

DDR .....Dual Data Rate

DUT ..... Device Under Test

DWM .....Dual Wound Machine

EMT .....Electro-Magnetics-Thermal

ESR ..... Equivalent Series Resistance

FIFO .....First-In-First-Out

FPGA .....Field Programmable Gate Array

FSM..... Finite State Machine

FWFT .....First Word Falls Through

HIL..... Hardware-In-the-Loop

HLS ..... High Level Synthesis

HMB .....Hardware Manager Board

IM..... Induction Machine

ITM ..... Ideal Transformer Model

KCL.....Kirchhoff Current Laws

KVL ..... Kirchhoff Voltage Laws

LB-LMC .....Latency-Based Linear Multi-step Compound

LSFR .....Linear Feedback Shift Register

LVDS .....Low Voltage Differential Signal

LU .....Lower-Upper

MMC..... Modular Multilevel Converter

NA..... Not Assigned or not mentioned or not present

ND..... Nodal Decomposition

NPM..... Norton Port Model

ORTiS ..... Open Real-Time Simulation

PCIe..... Peripheral Component Interconnect express

PEPDS.....Power Electronics based Power Distribution System

PGM..... Power Generation Module

PSC ..... Phase Shifted Control

PSM.....Phase Shifted Modulation

PWM ..... Pulsed Width Modulation

RC ..... Resistive Companion

RX..... Receiving

RT .....Real Time

SM.....Sub Module

SoC..... System-On-Chip

SOP .....Sum Of Product

SZS..... Ship Zonal System

SPST ..... Single-Pole-Single-Throw

TX ..... Transmitting

US+ ..... Virtex Ultrascale+ VC118

VB ..... Voltage-Balancing

CHAPTER 1

INTRODUCTION

Recent advances in Power Electronics and Power Distribution systems (PEPDS) are leading to an increasing complexity in converters architectures with their controls and to the employment of new SiC electronic switching devices which allows high switching frequency (~ 100kHz). Hence the power system design stage has increasingly become challenging as well as the testing, leading to higher risk of failure of the real system. In this sense, the HIL technique is the most common adopted solution both in industry and academia.

The employment of RT-HIL testing/simulation technique can facilitate the handling of system complexity. The realization of PEPDS simulation models either using GUI based commercial tools such as Simulink either coding them, for example in C++, it is for sure more practical and quicker then assembling the real hardware. The test execution is safer and faster and the data acquisition is simpler because there is no need of installing physical probes on a software model. Furthermore, PEPDS models development can be accompanied and supported by HIL integration tests.

Although HIL is a widely used and well-known technique there are still large margin of improvements related to the development of new simulation methodologies to obtain more accurate, scalable, and flexible simulation platforms specially for system level testing where the size of the system and the minimum achievable simulation time step are

two counteracting fundamental factors affecting the simulation performances in terms of accuracy, scalability, and flexibility. Those factors will be emphasized later in the work described in the next chapters.

1.1 Scope, purpose, and goals of this dissertation research work

This dissertation research work is focused on RT-HIL simulations for PEPDS and It is motivated by the aforementioned possibility of improving the HIL technique, with the purpose of developing RT-HIL methods capable of achieving high level of accuracy in a scalable and flexible way. The realization of such methods involves the creation of specific simulation platforms. In practice the proposed work aims to:

- realize a methodology for simulating in RT a SZS on FPGA applying the LB-LMC method to optimize resource usage and achieve small time steps
- show an example of this method for a CHIL simulation where the controller is TI-Delfino and a microgrid runs on a single Virtex7 FPGA
- show an example for a CHIL simulation involving multi-FPGA controller and MMC
- model the SZS converters (DAB and MMC) used for the simulation with state space method and with switching functions
- Realize communication interfaces for RT-HIL applications including co-simulation.

1.2 Background

In this section are presented some basic background notions about RT-HIL simulations with focus on the procedure adopted for this work.

1.2.1 RT and HIL simulations

Real time simulations consist in computer models of real systems running on digital platforms performing tasks with same time rate of the real counterpart, following its same wall clock. Hence the simulation time step adopted should be fixed as time moves forward in equal duration as the real wall clock time.

In Figure 1.1 a schematic summarizes the previous concepts and the basic structure of a RT simulation. A model, based on state space equations in the specific case of this work, is derived studying the behavior of the real system and its digital implementation is deployed on the simulation platform which can be either commercial (i.e., Opal RT) or custom (i.e., based on FPGAs, CPUs, etc.). The simulated system -the plant- and eventual simulated controls runs on the same platform. The time required to solve at each fixed simulation time step the model equations must be less or equal than the simulation time step and if this time interval is exceeded an overrun is detected which is not acceptable for a RT behavior. Model variables and outputs must be generated within the same length of time than the real system would.

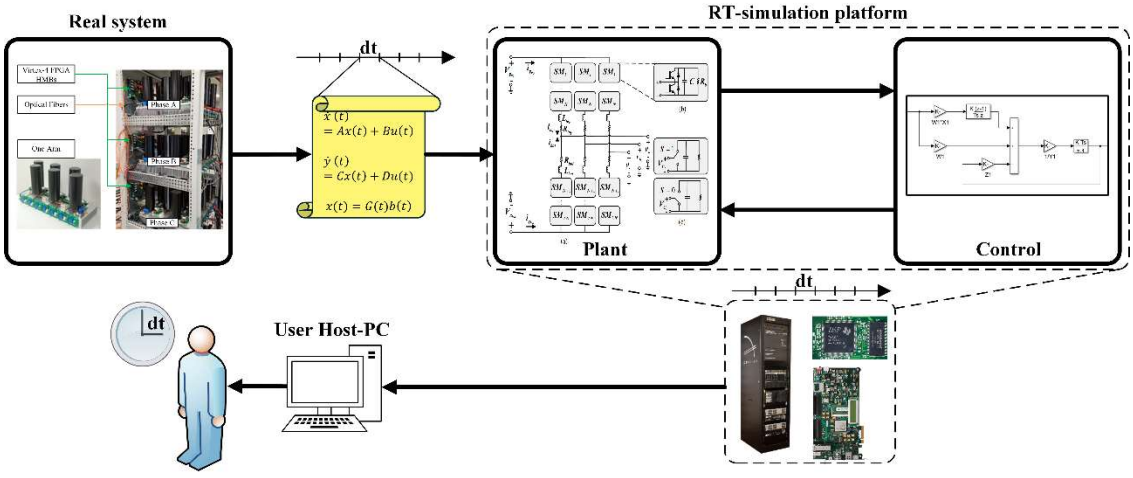


Figure 1.1 Schematic of the basic PEPDS RT simulation concept envisioned for this work.

Configuring models to run in RT allows to apply the Hardware in the Loop (HIL) testing technique to test components, also called DUT, of the real system by connecting them to the simulation platform where the remaining part of the system, the plant, is simulated. For the simulator and the DUT to interact with each other a communication interface is required as shown in Figure 1.2 which summarizes the basic structure of a HIL simulations. Normally the DUT is a control unit which is an embedded system based on programmable computational logic devices (i.e., CPU, FPGA, DSP, etc) used for control purposes. In this case the testing technique is defined CHIL.

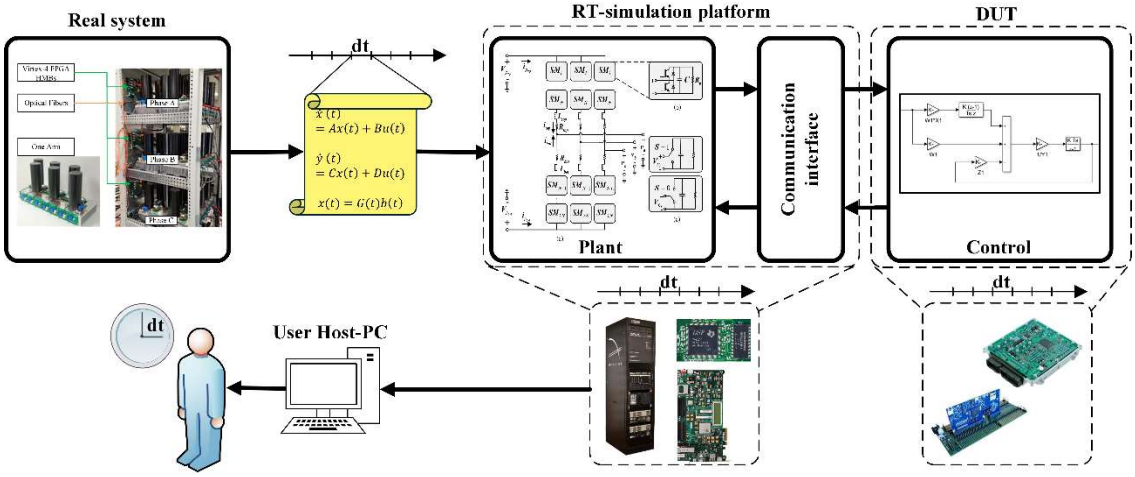


Figure 1.2 Schematic of the basic PEPDS HIL testing concept.

A RT simulation platform may require more communication interfaces in addition to the previous mentioned above.

For example, if the simulation model is decoupled over multiple devices, they will require communication interfaces to interact with each other. Similarly, these will be required to connect multiple control units, as summarized in Figure 1.3. RT and HIL simulation can be classified according to the type of execution as single rate or multi rate.

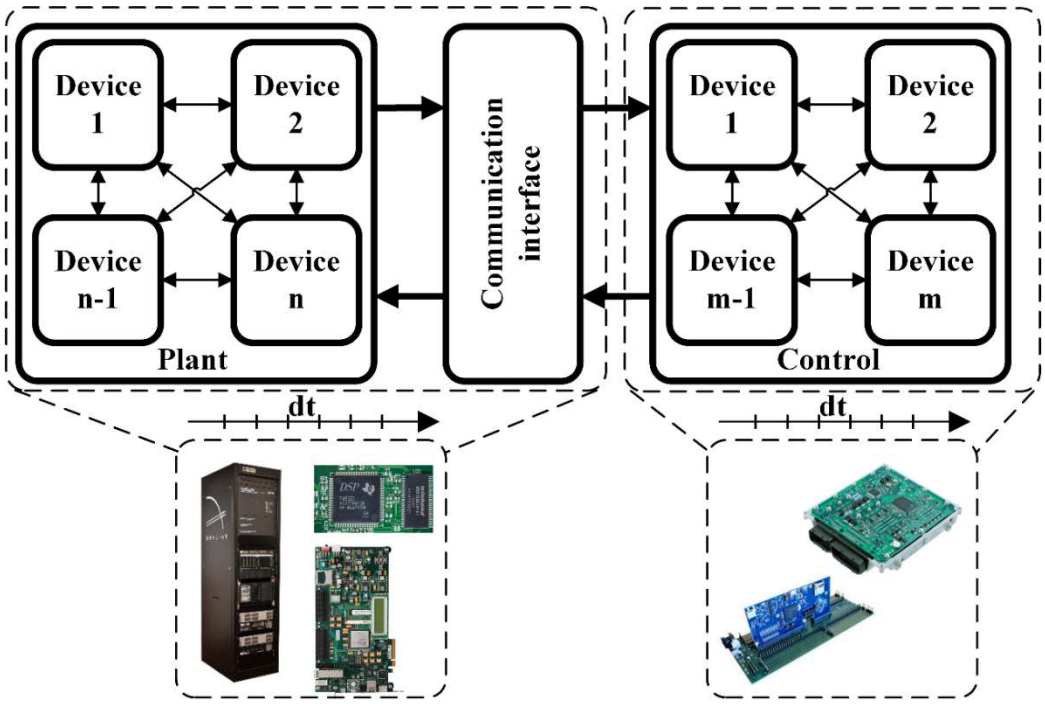


Figure 1.3 A schematic showing possible applications of communication interfaces within a HIL simulation platform.

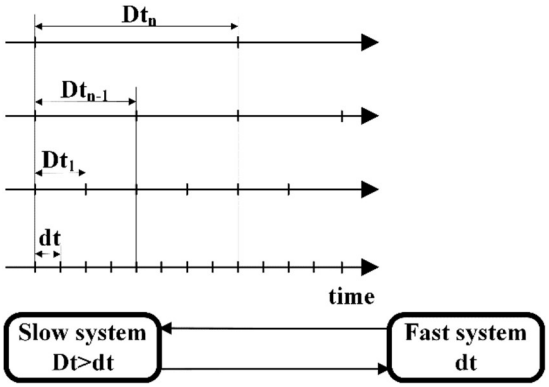


Figure 1.4 The multi-rate concept.

In the first case all the system is simulated with the same time step while in the second case different portions of the system (i.e., subsystems) are simulated with different time steps as illustrated in Figure 1.4 showing two subsystems being simulated with time steps DT greater then dt. The simulation is still happening in RT because the time rate considered is the same as the wall clock of the real system, but each subsystem is executed

with a different time interval. So, it is possible to identify a fast and a slow subsystem which communication needs to handle applying proper multi rate strategies which will be discussed further in this document.

1.2.2 Factors affecting simulation performances.

In this work, performances of simulation platform are evaluated in terms of accuracy, scalability, and flexibility.

The accuracy grows with time resolution and in decoupled systems depends as well on the communication latency introduced by the decoupling interface and the communication channel. The execution type (multi-rate or single rate) also affects the accuracy. For example, some simulators, to minimize the computational resource usage may require employing a multi-rate execution to simulate some part of the system with lower resolution (larger time step) which consequently reduces the number of samples (amount of information) hence the resemblance of the simulation with the real behavior.

The scalability, which is the ability of a system to handle an increasing number of tasks, depends on the maximum size of the system which the simulator is capable of simulating. It is affected by the platform interfacing which determines how many signals the simulator exchanges during communication with external devices. Time resolution also affect the scalability: for a given system size, increasing the required resolution (reducing the simulation time step) leads to an increased computational resource usage which can exceed the simulator device available resources preventing to simulate a system with a given size.

The flexibility relates to the ability of the system in adapting to different operating conditions which can be for example, the different simulated system size, different types

of interfaces, different execution type (single or multi rate) and different time resolution in the sense that the simulator is more flexible if can employ a wider range of simulation time steps. Hence the flexibility is a general concept embracing different simulation aspects.

1.3 Need for method that supports small time steps but also ensures scalability and flexibility

Considering the main notions given in the previous background, in this section it is further clarified the motivation behind this research which is strictly related to the need for a method that supports small time steps but also ensures scalability and flexibility.

As per their nature, PEPDS involve different level of complexity. On large scale, at system level, it is related to the grid topology and slow dynamic power flow phenomenon and on a smaller scale, at device level it involves the faster internal Power Electronics device behavior.

The system level testing is undoubtedly a practical and efficient simulation solution which in comparison with device level testing presents the advantage of exposing the entire simulated system behavior with the disadvantage of loss in accuracy due to computational limitation for increasing system size. Normally the study of detailed features is conducted at device level simulation where smaller simulation time steps, allowing higher accuracy, can be achieved but only for a limited number of simulated devices.

Without computational limitations, a simulation would be capable of exposing the entire system behavior as it happens at system level with same accuracy of device level simulations.

A RT-CHIL method capable of supporting at system level small simulation time steps ensuring at the same time scalability and flexibility would undoubtedly increase the

potential of the HIL technique. It would allow to realize more realistic and reliable testing tools reducing dramatically the risk of failure.

From the scientific point of view, the method would demonstrate that within the limits of available computational resources, an increasing level of accuracy can be achieved in scalable and flexible way. For example, keeping the same high level of accuracy, with small time steps (<100ns) with increasing system size and the simulator still flexible enough to adapt to different operating conditions such as being able to interact with other devices.

1.4 Hardware-in-the-Loop interfaces for testing several controllers in a scalable way.

As depicted in Figure 1.5, the HIL technique requires the DUT, to be connected to the test bench representing the plant (containing the simulated PEPDS to be controlled).

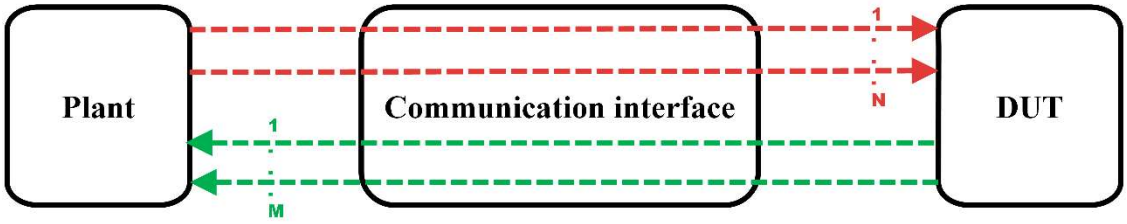


Figure 1.5 Schematic representing the basic HIL simulator structure.

Consequently, an appropriate communication interface must be implemented to allow the two devices to exchange data and work properly to reproduce accurately the real behavior of the system.

Considering the most common case found in literature -i.e. the CHIL-, the plant which is for example a simulated PEPDS will bring to the interface N signal buses, one per each PEPDS component requiring control (converters, transformers, IM, etc.) . Similarly, the DUT, which will be composed by controllers, generally one per each PEPDS

component requiring control, will have one signal bus per control unit for a total of M buses brought to the interface. The whole system will require to exchange signals over those M+N signals buses through the interface.

Consequently, the scalability is an important requirement for an HIL interface since its design will be affected by the size of the systems involved. Furthermore, it must be flexible enough to allow on the same simulator platform different simulation scenarios (different plants, different DUT, different number of signals, etc.).

Simulation interfaces can have different degrees of flexibility, for example the one used in [1] which connects a TI-Delfino Microcontroller to a Micro grid running on Virtex-7 is rigid since it is completely analogic. There is no protocol managing the data exchange and the number of communicable signals is limited by the number of available SMA ports on each board so there is one port per signal. The main limitation of that work was that no more than one controller unit could be host on the platform.

Similar limitation with a more complex interface based on Ethercat with analog I/O cable based connections involving a Ship Zonal System can be found in [2] and a similar work in [3]. In [4] is proposed a Fortran script based interface for laptop simulator for a 174 nodes feeder . In [5] is adopted an analog IO board for a Boost-Buck converter system with Ultra-capacitor. Recent advances in telecommunications and computer science are leading to the employment of high performance computational and storage capabilities and increased communication speed for which fiber optics-based communication interfaces are more suitable. To support this new type of solution, Xilinx has recently introduced the Aurora protocol in two different encoding versions, 8bit, namely Aurora_8B10B and 64bit, namely Aurora_64B66B.

As defined in [6] and in [7]:

"The Aurora 8B/10B protocol is a scalable, lightweight, link-layer protocol that can be used to move data point-to-point across one or more high-speed serial lanes."

The Aurora based interfaces are normally highly flexible and scalable since the communication happens through one fiber optic cable connection through which it is possible to exchange frame composed of hundreds of signals overcoming the scalability limitations of the analogic interfaces. It allows speed (line rate) of 10GBps using up to 10 communication lanes of the dedicated transceivers. Indeed many works found on literature [8][9][10][11][12] who aims to simulate large systems adopt Aurora based interfaces exploiting its high scalable features which allows to easily build a simulator without the cumbersome and tedious work of assembling a huge analogic cable based hardware interface.

The use of FPGA in the field of RT-HIL simulation is leading to more complex simulation platform where a CPU unit, taking care of the slowest system dynamics, due to its sequential operational nature, is interfaced through PCIe with an FPGA based unit taking care of the fast dynamics parts of the systems. The employment of high speed serial interfaces communication is becoming more very common [13][14][5][15][10][16][17].

1.5 Literature review of methods for RT simulations and HIL for PEPDS

Pure RT system level simulations employing time steps greater or equal than 50μs are found in [13] involving 101 levels MMC model simulated applying the state space Nodal method (SSN) described in [2] on an OPAL-RTCPU based simulator and [18], on a multi-FPGA based simulator. Other RT Multi Rate simulations [19] [14] with time steps greater than 50us down to 0.5us involves PEPDS with MMCs and switching frequencies

of 3000Hz. The simulation strategy is complicated since relays on elaborated simulator layout with CPU and FPGA, while in the other cases on a Xilinx Multiprocessing system. The models are detailed with look up tables and datasheets such as [19] to try replicating the switching behavior of some IGBT to perform device level analysis but being datasheet-based the simulator does not perform real switching actions. Instead, it elaborates look up tables data to interpolates temperature and voltage curves.

Regarding HIL simulation works, some are single rate simulations based on commercial CPU/FPGA simulators employing PCIe, using fiber optics as physical connector for the HIL interface, and involving MMC AC/DC grids. The dynamics involved are slow. Specifically in [20] it is simulated an AC grid with large time steps (between 50us and 0.5us) with switching frequencies of 230Hz. A simulation with similar characteristics is conducted in [21] for an MMC-BESS system involving switching frequencies of 10kHz . While a faster simulation, below 0.5us, is realized in [13] for a 401 levels MMC.

More complex HIL experiments are performed in multi rate fashion required for device level and EMT analysis. This implies decoupling the system in subsystem running at different time steps. The decoupling strategy may introduce communication delays such as in the case of the Stub line technique or the ITM. In [15][10][16][17] are used commercial simulators such as Opal which exploiting the joint capabilities of CPU and FPGA (CPU primarily used to run the controls) simulates large complex power systems in most of the cases including MMCs. The simulation time steps values spans from beyond 50us to few microseconds, remaining around the 500ns limit for device level simulations of IGBT devices which models are not full switching but based on datasheet with curve

fitting models. This happens as well in [8][9] except the fact that they employ a simulator based on US+ SoC processing unit and works like [11][12] which perform EMT analysis on power system involving Induction Machines and MMCs on a Virtex-7 FPGA.

In [22] an FPGA based simulator is capable of simulating a 401 levels MMC on FGPA with a small time step of 100ns. The model has been discretized using an implicit integration method, the Trapezoidal rule, which does not allow full parallelization of the computation leaving open margin of improvements which have instead been achieved in other works thanks to the employment of LBLMC method described in [23] for which some applications have already been developed for a SZS on Virtex-7 FPGA using as HIL interface and an analog IO custom board and in a work which is currently under publication process where an MMC with AC and DC links has been simulated connected through fiber optics, using the Aurora protocol, to a multi FPGA ring based control.

Complex simulation tasks may require more than one computational unit when resources are not enough to achieve the simulation goals (i.e., to run at a given simulation time step, to simulate a system of a certain size, etc.) and decoupling the system according to the employed technique may introduce communication delays which makes the simulation an approximation of the real solution.

In [24] it has been introduced a new decoupling technique, the Nodal Decomposition, based on Norton-Port equivalent models which extends the capabilities of the LB-LMC method to simulated large systems on multiple FPGAs with a decoupling interface without introducing any delays.

Theoretically the solution of the decoupled system, differently from the case of the ITM is the same solution of the original systems. The practical simulation of the system on

the multi-FPGA simulator is affected by the trace delay introduced by the physical communication channel connecting the two devices.

The main idea behind the purpose of the proposed dissertation work is to introduce a new RT-HIL methodology which exploiting the nodal decomposition and LB-LMC capabilities allows to simulate with a small time steps (<100ns) a complex PEPDS such as the SZS, including two MMC rectifiers, decoupled over three US+ FPGAs, and which communicates through the Aurora protocol based Interface to a slower dynamic PGM, composed by the Gas Turbine and a Dual Wound Machine (DWM), running on OP5607, a CPU based OPAL-RT test bench. The overall system including the multi-FPGA simulator platform and the OPAL-RT testbench is a Co-Simulator platform being the union of two platforms, specifically one commercial and slow providing easy of implementation and another custom and fast but presenting more complexities.

1.6 Conclusions

Recent advances in microgrid technologies are leading to more complex PEPDS and to reduce the risk of failure, very accurate RT-HIL simulation methods are required. While the HIL is a common adopted testing technique for PEPDS, as shown by the literature review presented in this chapter, there are still margin of improvements for state-of-the-art system level testing which can be further improved to achieve a higher level of accuracy -employing small time steps- when simulating large systems involving fast dynamics (i.e., microgrid with fast switching converters).

This work aims to improve state-of-art RT-HIL methodologies by proposing new simulation methods based on the LB-LMC and nodal decomposition with the introduction of newly developed communication interfaces to facilitate the simulation.

CHAPTER 2

DAB AND MMC CONVERTER MODELS FOR REAL TIME AND HIL SIMULATIONS

This chapter introduces one of the main contributions of this dissertation work, specifically, the implementation of DAB and MMC converters models for RT-HIL simulations, and the demonstration that they fit the LB-LMC implementation with small time steps for RT simulations.

2.1 Review of LB-LMC

The LB-LMC method is a highly parallelizable simulation method designed for RT simulation of dynamic electrical or multi-physics systems. In this section, we provide a summary description of this method which is detailed in [23]

This method is derived from the RC method, or Electro-Magnetic Transients Program (EMTP) method, solving dynamic systems as a set of linear equations Gx=b every simulation time step, where G is the conductance of the system, b is the current contributions of components, and x is the node voltages of the system. Unlike traditional RC method, the LB-LMC method models all nonlinear components in a linear network as functional voltage sources with series resistance or as current sources with parallel conductance.

These series resistances or parallel conductance are held fixed and are inserted into the *G* conductance matrix to stay with standard form of RC components.

The nonlinear behavior of the nonlinear components is then reflected in the voltage or current source that is updated every simulation step through an internal step that computes the state equation of the component to update said source. The nonlinear component state equations are expressed as:

$$\frac{di_i^n}{dt} = f(v, i, x_i^n, u_i^n, t) \tag{1}$$

$$\frac{dv_i^n}{dt} = f(v, i, x_i^n, u_i^n, t) \tag{2}$$

Where v is the vector of the network node voltages, i is the vector of the network branch currents, x_i^n is the vector of the state variable internal to the i-th nonlinear component, and u_i^n is the vector of the input internal to the i-th nonlinear component. Components with multiple terminals can be described by a mix of these current and voltage sources. These equations are explicitly discretized to obtain:

$$I_i^n = f(v(k), i(k), x_i^n(k), u_i^n(k), k) \tag{3}$$

$$V_j^n = f(v(k), i(k), x_i^n(k), u_i^n(k), k) \tag{4}$$

Since the state equations for I^n_i and V^n_j are explicitly discretized and only depend on the solutions from previous time step, and the equations are independent from one another, each nonlinear component can perform its internal step in parallel to other components.

From these state equations, the source contribution vector b can be updated and the system solution each time step can be found with:

$$Gx(k+1) = b(v(k), i(k), I^n(k), V^n(k), k) \tag{5}$$

From having the conductance matrix G held constant due to consisting of only fixed conductance, matrix inversion or LU factorization for the LB-LMC method system solver

can be performed offline. Only forward and backward substitution to solve the system is performed each time step.

In Figure 2.1, it is shown the solution flow for LB-LMC.

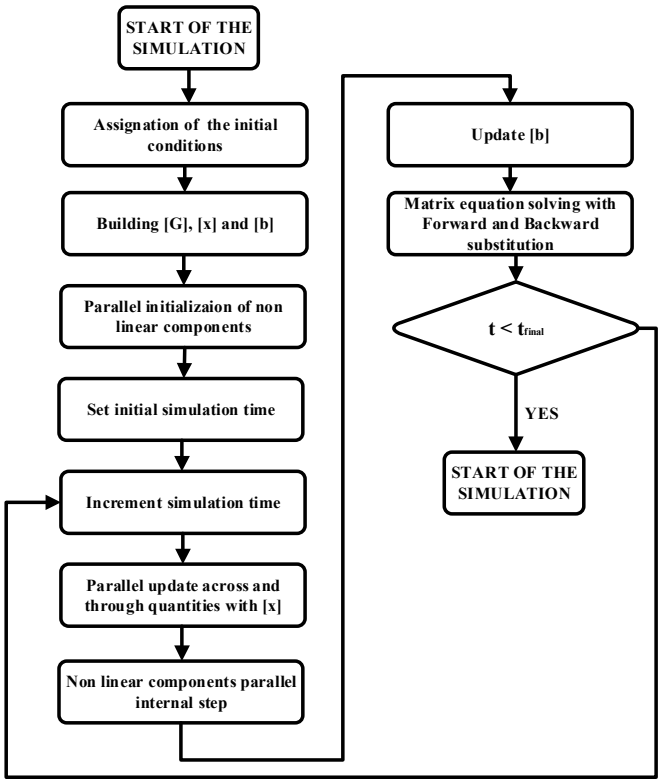


Figure 2.1 LB-LMC Solution Flow.

In this flow, G, x, and b are built from initial conditions and the LU factorization of the conductance matrix is performed. Once all nonlinear components are initialized, the simulation loop begins. Each iteration consists of each component performing its own internal step in parallel, then the source vector b is updated. From the updated b vector, the system solution *x* is computed via forward and backward substitution and saved for the next step. The simulation loop continues until final simulation time is reached.

The most important characteristic of the proposed approach is the use of explicit integration for the nonlinear components, while the linear part of the network is integrated

using an implicit numerical method. While the use of a linear multi-step compound method offers always better accuracy and stability property than the worst of the integration methods used, at the same time the use of an explicit integration method algorithm always implies some concern related to stability and accuracy which have been deeply discussed and analyzed in [23].

2.1.1 Simulation Engine

In this section, we discuss how the LB-LMC method is encapsulated and implemented for FPGA, real-time execution. Further discussion of this topic is found in [1].

2.1.1.1 Component entities

For each nonlinear component type used to model a system, a FPGA entity is developed. As input, these component entities take the system solution computed in a previous time step. Along with system solution, component entities can also take other input signals to control behavior of the entity, such as switch control signals for a DC/AC converter component. At the beginning of each time step, the component entities sample and register their inputs. From these inputs and past internal states, the components perform their internal step for (3) and/or (4) and compute their source contributions.

The component entities perform computational operations for their internal step in a non-pipelined, dataflow (data-driven) manner. In this manner, all internal steps operations are immediately executed in response to any changes in the component entity inputs, or in the results passed between operations. Moreover, these operations are all performed in parallel to one another, no matter the dependency between operations. Due to this execution flow, internal step operations never wait on prerequisite operations to

finish to begin their own execution, the operations converging to correct results as prerequisite ones complete. All internal step operations of a component are expected to complete with correct results within a single pass before new inputs are registered.

2.1.1.2 System solver

A dedicated system solver FPGA entity is created to compute the system solution. This solver entity takes as input the component source contributions and accumulates these contributions together to create the whole source vector b used to compute the system solution. The entity provides the system solution vector *x* as output which are fed back to component entities as input for the next time step execution.

Unlike the original LB-LMC method, the system solver entity in the FPGA implementation does not use forward-backward substitution for system solution computation. Instead, this entity uses an inverted conductance matrix precomputed offline and multiple algebraic sums of product (SOP) expressions to find the system solution. In this approach, the system solution is found by solving (5) for the vector *x* like in (6), where *A* is the inverted *G* conductance matrix ($A = G^{-1}$).

$$x = Ab \tag{6}$$

This solution is computed by expanding the multiplication between A and b matrices into SOP expressions, like seen in (7), which are to be each computed individually from one another. Since the inverted conductance matrix is fixed, the A terms in the SOP expressions can be defined as constants in said expressions.

Similarly, to the component entities, the system solver uses a dataflow approach to perform its computations in the FPGA implementation. In this approach, the system solver solves all its SOP solution equations entirely in parallel using a data-driven manner.

$$x = Ab = \begin{matrix} A_{11}b_1 + A_{12}b_2 + \cdots + A_{1n}b_n \\ A_{21}b_1 + A_{22}b_2 + \cdots + A_{2n}b_n \\ \vdots \\ A_{n1}b_1 + A_{n2}b_2 + \cdots + A_{nn}b_n \end{matrix} \quad (7)$$

All solution equations are each given dedicated computational units, composed of combinational multiplier and adder units on the host FPGA, which operate independently from each other. As component source contributions are provided to the system solver, the computational units will compute the system solutions immediately without the need to wait on any control or clock signal to initiate computations; only the input contributions are required to start computation. This approach allows solutions to be produced without delays induced from performed clocked operations sequentially.

2.1.1.3 Engine composition

To perform simulation of a system with the FPGA implementation of LB-LMC method, a simulation engine like seen in Figure 2.2 is composed, consisting of multiple component entities and one system solver entity tailored to the system simulated.

In the engine, a component entity for each nonlinear component of the system is instanced and their source contribution outputs are linked to the appropriate inputs of the instanced solver entity.

The system solution output of the system solver is fed back to the component entities inputs, the components taking solution elements that correspond to their model terminals.

If component entities require input from peripherals such as a switch controller, the appropriate FPGA elements are added to the design and linked to the requiring component entities.

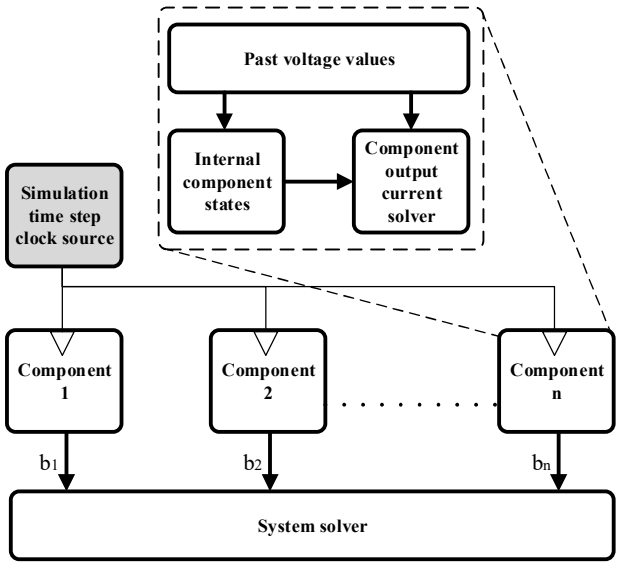


Figure 2.2 Simulation Engine structure.

2.1.1.4 Execution scheduling

The execution scheduling of the simulation engine was set to use a single pass, dataflow approach. In this approach, the simulation engine execution is performed in a single pass, bounded to a system clock whose period is equal to the simulation time step. On the start of the time step, the component entities sample their inputs for the system solution from past time step and any peripheral inputs on the host FPGA. Then, the components perform their computations. As source contributions' values are computed, the dataflow system solver will immediately compute the current time step system solution without wait. The choice of time step clock period is selected to be greater than the computational time needed by the simulation engine for stable operation.

2.1.1.5 Numerical representation

So that computational delays for the simulation engine are reduced and mostly dependent on the low propagation delays of the FPGA primitives, fixed-point arithmetic logic is used instead of floating-point for all computations. Common floating-point

arithmetic implementations, such as IEEE 754, typically require complex, high-latency, pipelined operations to handle their sophisticated formats. Fixed-point arithmetic logic, on the other hand, can be easily created with simpler combinational logic for integer arithmetic which does not require pipelining. Due to not needing to be clocked or pipelined to produce an output, fixed-point computational delay can depend solely on propagation delay of the comprised logic primitives. Since fixed-point arithmetic hardware is much simpler than floating point hardware, these propagation delays can be kept low. Moreover, many FPGA platforms have built-in integer DSP slices or blocks which can be applied to accelerate operations and reduce delays and resource usage of integer and fixed-point arithmetic. The main downside to using fixed-point arithmetic is limited numerical precision compared to floating point, which can adversely affect numerical stability and accuracy. However, this limitation can be alleviated with careful selection of integral and fractional bit widths for fixed-point signals within the simulation engine, giving numerical accuracy comparable to use of floating-point arithmetic.

2.1.1.6 External interfacing

To allow the simulation engine to interface with external devices separate from the FPGA, such as switch controllers and analog/digital converters, for HIL simulation, extra logic is needed for this interfacing. For each external input and output signal of the simulation engine, simple registers are added to allow the signals to be held constant through a time step. Every time step, these registers are clocked to sample corresponding input or output signal for the simulation engine. When the simulation engine begins execution for the start of a time step, the simulation engine will read from the input registers. Similarly, external devices that read the output of the simulation engine will read

from the output registers of the engine at start of each time step. Generally, the output of the simulation engine will be states (currents or voltages) of the modeled system stored in fixed-point format. For external devices reading these outputs, said outputs will often need to be normalized for the input value range and numerical representation of the external device in question. For example, a digital/analog converter may expect the values from the simulation engine to be positive-only integer values which would require the fixed-point values of the engine to be normalized to such range of values.

2.2 MMC model, scalability, and accuracy

The MMC is one of the most attractive electrical devices both for utility and high frequency applications (automotive, naval, or modern microgrids) due to its flexibility, modularity and reduced needs of passive elements with respect to other types of converters with consequent savings in terms of space and costs of components. On the other side its topology and principle of operation requires the adoption of complex control algorithms and a considerable effort in testing the device before production to reduce the risk of failure and damages.

In the following sections it is described the full switching state space implementation of a half-bridge MMC which has been tested individually on small micro grid single-converter network.

2.2.2 LB-LMC MMC model

In the LB-LMC framework the MMC is represented as a state space model with five current-type ports -eq. (1) and (3)- in the solver nodal part. Several state space model of MMCs have been developed in literature and the selection of the state variables depends upon the simulation needs. In [10] the arm currents were selected to be fed to a 5μs time

step sub module (SM) transient simulation. In [25] besides the capacitor voltages, DC currents, line currents and circulating currents are considered.

In [26], the state variables are selected to better represent the circuit dynamics to unveil unused switching states of the SMs. In [27] an averaged dynamic model is used to study large-signals terminals and capacitor dynamics.

The model formulation used in this paper selects as state variables the arm currents and capacitor voltages.

The developed MMC model includes a switching representation of the half bridge SMs. While this is based on a switching functions approach, we include switches conduction resistances, freewheeling diodes and a bleeding resistance connected in parallel with the module capacitor -this is included to match the hardware prototype used for validation, but it can be easily excluded for other studies.

Let us consider as reference the circuit in Figure 2.3 (a) with bipolar DC Bus link voltages $V_{g_{top}}$ and $V_{g_{bot}}$, and three phase output voltages v_j-subscript j is used to indicate phase a, b and c. Each phase leg has a top and a bottom arm, each one having a number of SMs equal to N.

The arm equivalent resistance and inductance are R_{top} and L_{top} for the top arm and R_{bot} and L_{bot} for bottom arm.

As indicated in Figure 2.3 (b) each SM is composed by two switching elements $S_1$and S_2 with diodes (d_1, d_2) in parallel with a capacitor C and a bleed resistance R_b to reflect the MMC converter used to validate the simulation model. $S_1$and S_2 are represented with a single pole single throw switch in series with its conduction resistance of value R_{on}.

The capacitor is modelled with a capacitance of value C in series with its ESR.

In the following model equations, each SM capacitor voltage is indicated as v_{Cij} where the subscript “i" indicates the upper arm (SMs i = 1 to N) and the lower arm (SMs i = (N+1) to 2N).

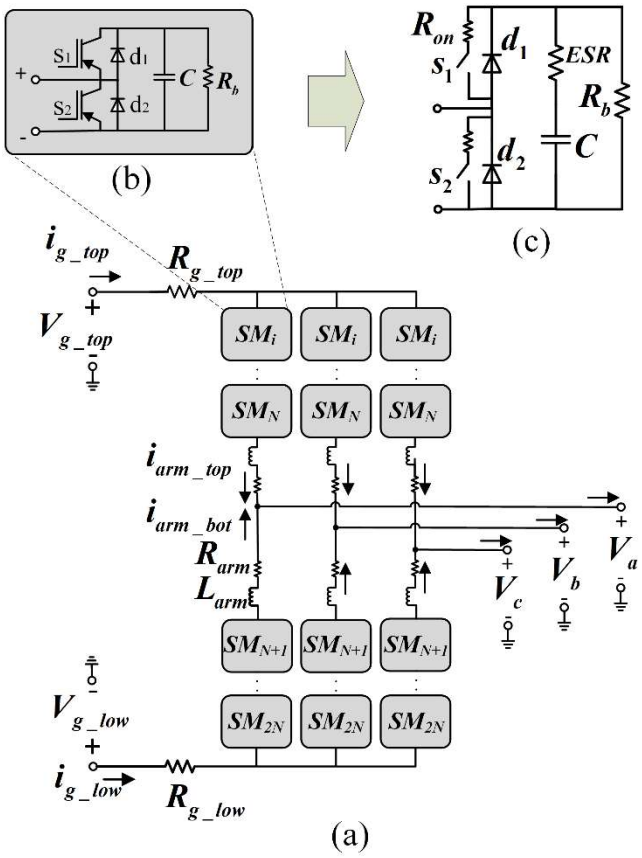


Figure 2.3 The MMC model, (a) the MMC circuit, (b) a detail of the SM circuit (c) the detailed model of SM.

The modes of operation considered are with the capacitor inserted or bypassed depending on the conduction state of the gates (s_1, s_2) or the diodes (d_1, d_2).

The state variables considered are the top and bottom arm inductance currents i_{top_j}, i_{bot_j} and the capacitor voltages v_{Cij}. Equations (8) and (9) are obtained by writing a KCL for each SM respectively of the upper and lower arm. In these equations, C_i is the SM capacitance, S_{ij} is the switching function. Depending on the gates values s_1 and s_2-, g_C

represents the conductance of the SM capacitor branch and R_b is the resistance value of the safety bleeding resistor of the real hardware MMC employed for model validation. The state-space equations (10) and (11), are derived from the KVL written for each independent loop from the DC link to the AC load for each phase leading. The S_{ij} can be equal to one or zero to respectively model the SM's states inserted and bypassed. As indicated in the equations below, S_{ij}depends on the gates states (s_1, s_2). Each gate can be either opened, hence its state will be equal to 0 or closed hence equal to 1. When the SM is inserted S_{ij}=1 then $(s_1 = 1 \,\&\, s_2 = 0)$and when it is bypassed S_{ij}=0 then $(s_1 = 0 \,\&\, s_2 = 1)$. During dead time state $(s_1 = 0 \,\&\, s_2 = 0)$ the value of S_{ij} is determined by the arm inductor current which according to its direction will flow into the proper diodes determining the SM's state.

$$C_i \frac{dv_{Cij}}{dt} = S_{ij}(s_1, s_2)\left(i_{top_j} \frac{R_b}{\frac{1}{g_C} + R_b} \right) - \left(1 - S_{ij}(s_1, s_2)\right)\left(\frac{v_{Cij}}{R_b}\right) \tag{8}$$

$$C_i \frac{dv_{Cij}}{dt} = S_{ij}(s_1, s_2)\left(i_{bot_j} \frac{R_b}{\frac{1}{g_C} + R_b} \right) - \left(1 - S_{ij}(s_1, s_2)\right)\left(\frac{v_{Cij}}{R_b}\right) \tag{9}$$

$$L_{top} * \frac{di_{top_j}}{dt} = V_{g_{top}} - R_{top} * i_{top_j} - \sum_{i=1}^{N} S_{ij}(s_1, s_2) * v_{cij} - v_j \tag{10}$$

$$L_{bot} * \frac{di_{bot_j}}{dt} = V_{g_{bot}} - R_{bot} * i_{bot_j} - \sum_{i=N+1}^{2N} S_{ij}(s_1, s_2) * v_{cij} - v_j \tag{11}$$

$$b_1 = i_{g_{top}} = i_{top_a} + i_{top_b} + i_{top_c} \tag{12}$$

$$b_2 = i_{g_{bot}} = i_{bot_a} + i_{bot_b} + i_{bot_c} \tag{13}$$

$$b_3 = i_{top_a} + i_{bot_a} \tag{14}$$

$$b_4 = i_{top_b} + i_{bot_b} \tag{15}$$

$$b_5 = i_{top_c} + i_{bot_c} \tag{16}$$

Applying the LB-LMC approach, the MMC model shown in Fig. 1 can be represented as an entity composed of five current-type ports, each one providing a current contribution b_k in parallel with a conductance g_k, with subscripts k=1,...5 corresponding to the component naming as shown in Figure 2.4. It is worth stressing that the developed MMC model is not an average model and the current sources of Figure 2.4 are updated at each time step using the equations (3)-(4) without any loss of information on the individual module switching behavior.

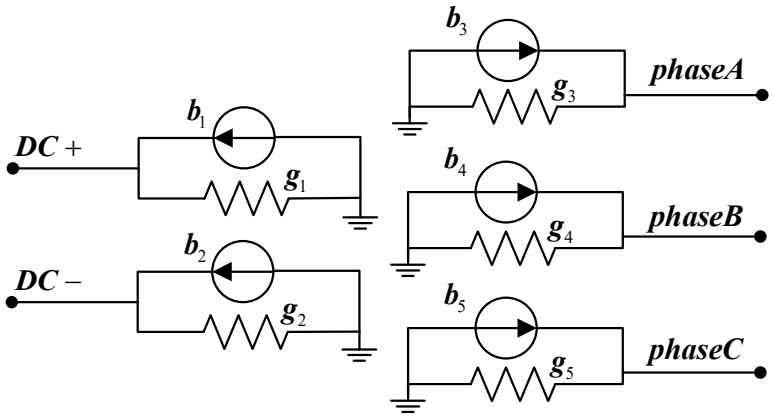


Figure 2.4 LB-LMC equivalent model of the MMC.

The equations (12) to (16) are used to compute the MMC equivalent LB-LMC model source contributions, b_1 to b_5, based on the top i_{top_j} and bottom i_{bot_j} arm currents computed by the state space equations (8) to (11). The contributions, b_1 and b_2 are equal to the DC link currents while b_3 to b_5 are equal to the phase currents i_j.

At each simulation time step, the MMC current contributions are computed and used to compute the system solution as in (5). The nodal voltages of the DC link and of the

three phase terminals v_j are then used at the next time step to update the state equations of the MMC converter. In equation (5)), *G* is the system conductance matrix, precomputed and inverted offline (i.e., the inversion computations are not executed on the FPGA); x is the vector of the nodal voltages and *b* is the vector of the currents contributions. The developed MMC model, together with other power electronics converter models, and the LB-LMC solver are distributed as an open-source package [28].

2.2.3 Accuracy

In this section it is described the MMC model validation by comparing the C-simulation results of such a model conducted on a commercial CPU based PC computer, with a real hardware MMC to show the accuracy of the developed model.

2.2.3.7 Hardware reference system

An MMC prototype, shown in Figure 2.5 is used for the model and HIL testing validation. The control hardware of the prototype is also used as Device Under Test (DUT) in the HIL testing. This MMC consists of six Half-Bridge (HB) SMs per arm. An SM is developed using the SEMIKRON SEMiX202GB066HDs HB IGBT module and the SEMIKRON SKYPER 32 PRO R IGBT gate driver triggers this IGBT module. Table 2.1. lists the key parameters.

The setup is designed for full load apparent power rating of 60k VA and the maximum DC bus voltage of 800 V. For flexibility purposes, the SM capacitance is 15 mF; the selection has been made based on market available capacitors' size, cost, and ESR.

Given the 15 mF of the SM capacitors and considering the peak magnitude of the double-fundamental-frequency circulating current, I_{2f} as 15 A the resulting value for arm inductors is 1 mH.

Each MMC phase is managed by a custom FPGA-based control platform, denoted as a Hardware Manager Board (HMB), which receives measurements from sensors and provides gate signals to each SM of its associated phase. Each HMB is composed of a Xilinx Virtex-4 XC4VFX20 FFG672 FPGA with four high-speed Multi-Gigabit Transceivers (MGT) and twenty analog-to-digital converters (ADC) channels. The ADCs are Texas Instruments (TI) ADS7863 which have four dual, 12-bit fully differential input channels grouped into two pairs for high-speed, simultaneous signal acquisition. Between HMBs, the three-phase ring communication is established using 8 bits of data width having 1.25 Gb/s line rate and a 125 MHz reference clock. The controller has a clock frequency CLK_C of 125 MHz. For all three phases, the control modulating signal *m(t)* is generated in the control master FPGA HMB-01 as depicted in Figure 2.6.

Table 2.1 MMC Prototype Hardware and AC System Parameters.

Symbol	Description	Value
P	Nominal Three phase active power	60 kW
V_{xn}	Nominal AC Phase RMS voltage	208 V
N	Number of SM per lower/upper arm	6
C	SM capacitance	15 mF
L_{top}, L_{bot}	Arm Inductance	1 mH
R_{top}, R_{bot}	Arm resistance	0.1 Ω
R_b	Capacitor bleed safety resistance	5.56 kΩ
V_{ci}	Max SM capacitor voltage	400 V
f_c	Carrier frequency	664 Hz
N_c	No. of carrier for PSC-PWM	6
f_{sw}	Effective Switching frequency	3.98 kHz
R_{on}	Swithcing ON resistance	1.0mΩ
ESR	SM'capacitor series resistance	0.113Ω

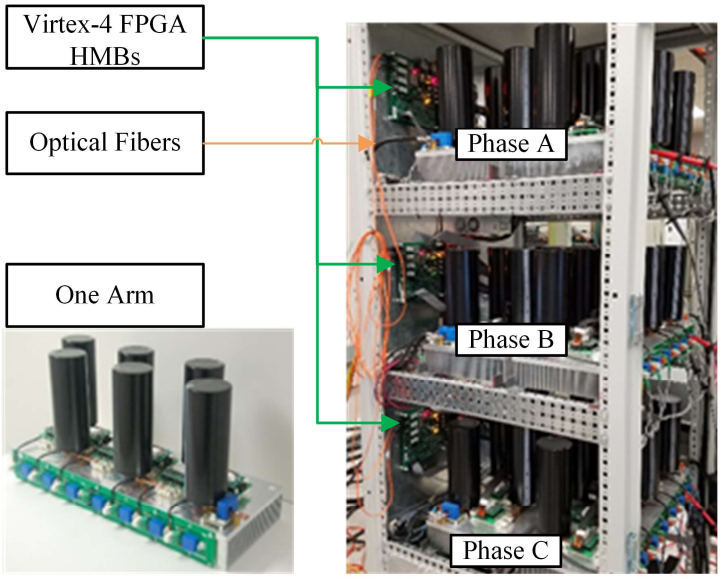


Figure 2.5 Three-phase MMC prototype hardware, (a) arm module and (b) three phase structure with Virtex-4 FPGA HMBs and optical fiber.

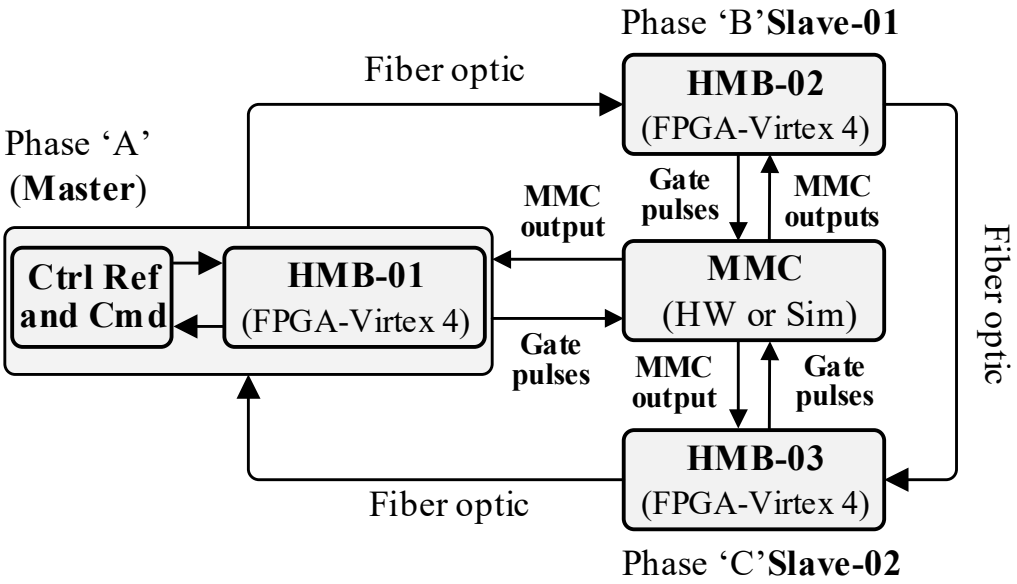


Figure 2.6 Block diagram of the control for both simulated and real Hardware (HW) MMC

HMB-01 also generates digital PWM gate signals and does the Voltage-Balancing (VB) for Phase A. HMB-02 and HMB-03 generates digital PWM gate signals and performs

VB for Phase B and C, respectively. These two-phase controllers (HMB-02 and HMB-03) receive *m(t)* from HMB-01 via high-speed optical fiber ring communication. The ring is a checksum-based custom communication protocol that is developed for data transfer between HMBs. In this work, the phase shifted carrier (PSC) pulse width modulation (PWM), detailed in [29], is used. Since, the MMC consists of six half-bridge SMs per arm, there are six 60^0 phase shifted carriers. The generation of the double updated modulating signal $m_s(t)$ and measurement sampling trigger are illustrated in Figure 2.7. Only the first triangular carrier is shown for clarity. The digital PWM gate signals are produced by comparing $m_s(t)$ and the carrier signal $A_{cr}(t)$. The switching frequency of the MMC, f_{sw}= Nf_c.

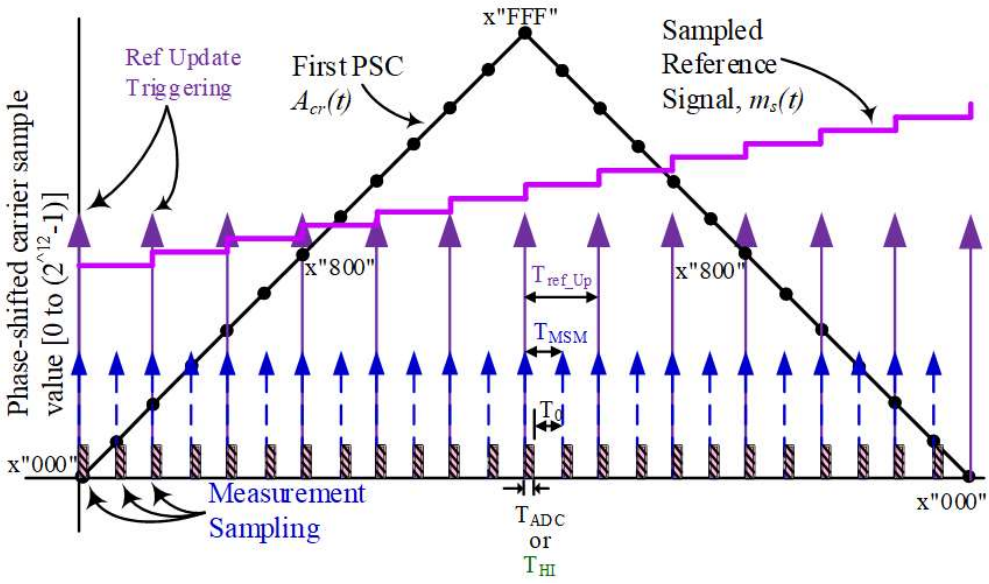


Figure 2.7 Generation of double updated modulating signal ms(t) and measurement sampling trigger

Since its double update control, thus the reference update period, T_{ref_up} is $T_{sw}/2$. The measurement sampling has a period, T_{MSM} is equal to $T_{ref\_up}/2$ or $T_{sw}/4$. The operation of the HMBs is only different in the measurement section with respect to connection of the

MMC hardware versus the simulator. For hardware experiments, when the measurement sampling event is up, it triggers the controller to receive new measurements from the MMC after T_{ADC} time, that is the HMB ADCs program and conversion time. For CHIL experiment, when a measurement sampling event is up, the control updates the measurements coming from the simulation engine after T_{HI} time that is total time for one cycle of the CHIL. During T_0 time the MMC measurements are stored in controller memory and the measurements are used for voltage balancing and protection of the system. The voltage balancing presented in [30] has been utilized for this work.

2.2.3.8 MMC Model Validation

The validation of simulation model is of particular importance in the evaluation of a simulation method. In this paper this is even more critical when we analyze the performance of the proposed Aurora interface.

Some reference work for validation of MMC models are reported here: [31][32][33]. To clarify the effectiveness of the presented work, a model validation has been conducted comparing the MMC phase and capacitor voltages and circulating currents waveforms with the ones obtained by operating a real MMC. The accuracy has been evaluated by computing the two-norm error.

The laboratory prototype MMC described in Section IV has been used to validate the MMC model and is referred here as HW. The model is executed on a CPU-based workstation, and it is referred to here as Csim.

For validation purposes, Csim and the real MMC implement the same control logic, a DC bus voltage of 620V and a resistive load power of 4.5 kW. The results have been compared by computing a two-norm error. The general formula for a p-order norm error

perr is presented in(17). Here p indicates the norm class is set equal to two since the Euclidean norm for the error calculation is considered.

The series of n reference samples indicated by subscript i=1,...,n, used to compute the error is $x_{ref\,i}$ while the series of measured samples for which the error is evaluated with respect to the reference is indicated as $x_{meas\,i}$.

$$\|perr\| = \sqrt[p]{\sum_{i=1}^{n} \left| \frac{x_{meas\,i} - x_{ref\,i}}{x_{ref}\ i} \right|^{p}} * 100 \tag{17}$$

The two-norm error of the phase voltage depicted in Figure 2.8(a) is equal to 8.7%. In this case, the results from the HW execution correspond to $x_{ref\,i}$ while the results from the Csim simulation correspond to $x_{meas\,i}$.

Differences in the phase voltages shown in Figure 2.8(a) are mainly due to circulating current differences between the hardware experiment and simulation as shown in Figure 2.8 (b).

This is mainly due to the mismatch of passive elements between the modules - capacitor and bleeding resistance. The results reported here are obtained my randomizing the values of these passive elements -considering the fabrication tolerance- around their nominal value, in this way we obtained a good matching without matching each individual parameter and so keeping the model of more general interest.

To demonstrate the free willing diode behavior -somehow unusual in switching function-based models- we analyze the dead time behavior of an individual module.

With reference to Figure 2.9 the quantities considered for the analysis are the voltage at the SM's terminals for a model without dead time $v_{sm}$and for the model with diodes and dead times v_{sm_ddt}.

The voltages should approach the capacitor voltage v_{c_ddt} when the SM is inserted and zero when the SM bypassed.

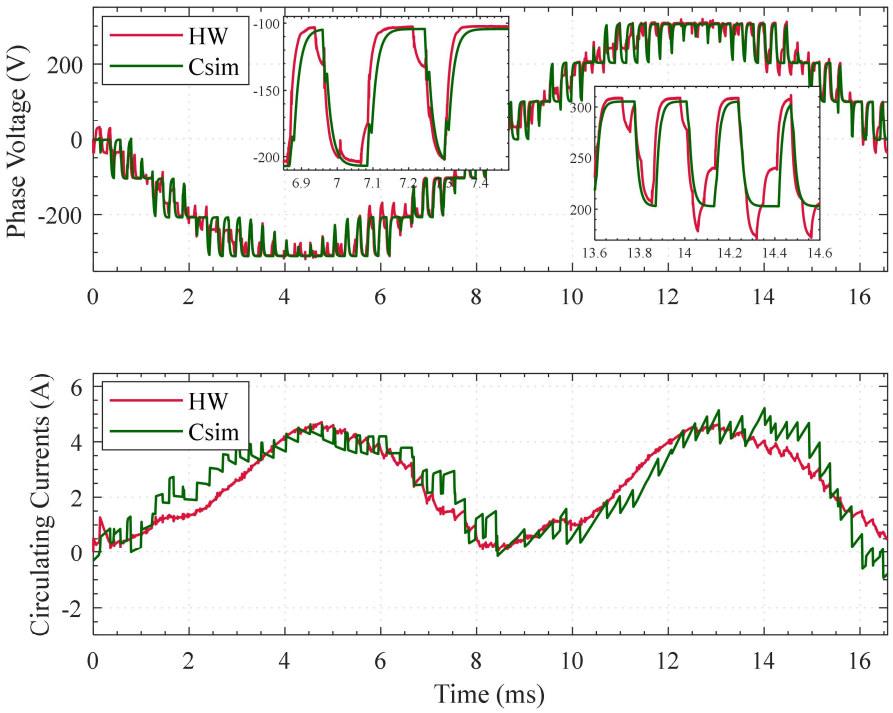


Figure 2.8 Hardware experimental and simulation results comparison (a) phase A voltage and (b) phase A circulating current.

The leg A top arm current i_{arm_top} is considered. When positive, it will flow through diode d_1 else diode d_2. In Figure 2.9 we reported the typical four conditions for analyzing dead time behavior of an half bridge topology, in Figure 2.9 (a) with the SM transitioning to inserted mode and positive current and Figure 2.9 (d) with the SM transitioning to bypassed mode and negative current the dead time effect is not visible since the diode conducting is in parallel to the switch conducting after the transition.

Vice versa the effect is well visible in Figure 2.9 (b) (c) where the diode conducting during the dead time is in parallel with the switch conducting before the transition. Those

conditions reflected on the SM circuit with corresponding arm current color are illustrated in Figure 2.9 (e).

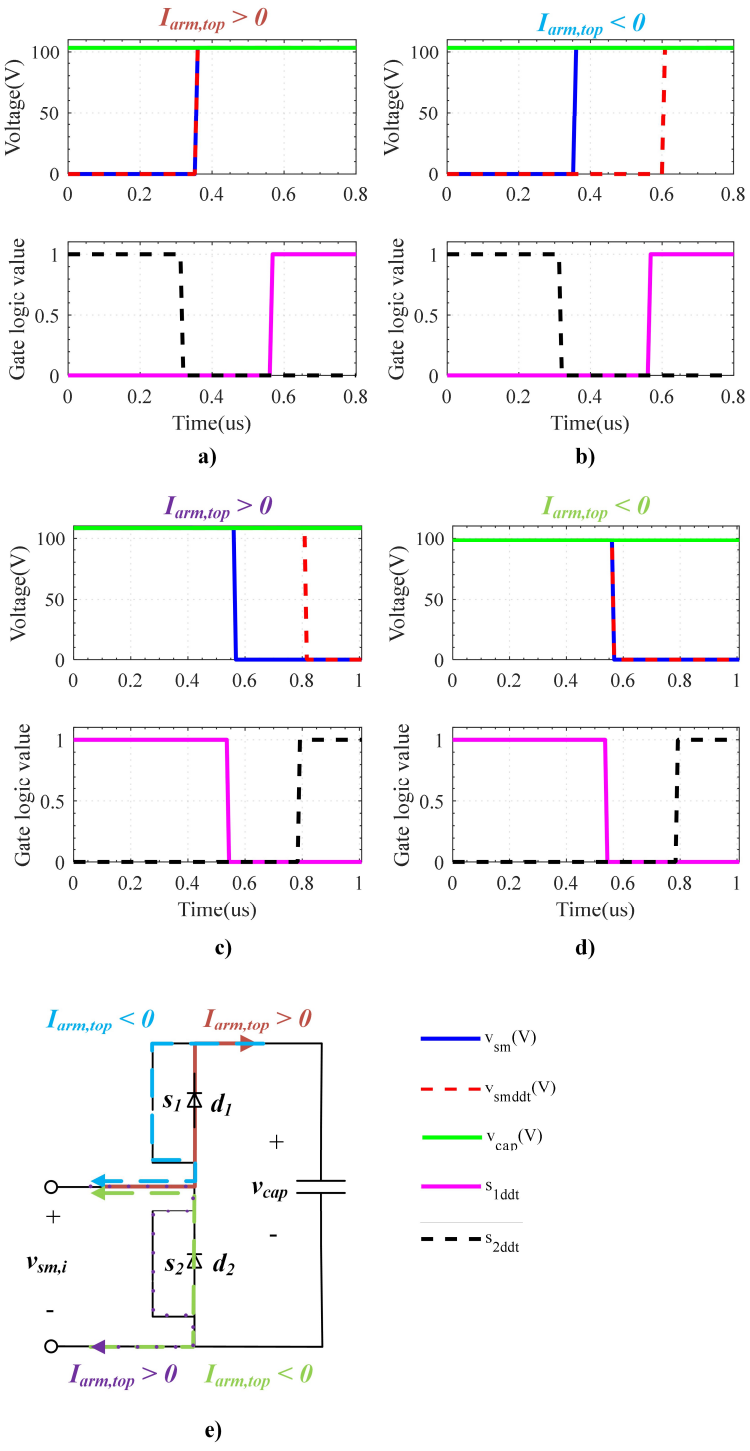


Figure 2.9 Switching dead times implementation.

To conclude the model validation in Figure 2.10 we compare the phase A 1st capacitor voltage computed by the simulation model with the one acquired from the MMC prototype. In this case rather than directly computing the two-norm error we calculate the percentage error – ErrM- by first removing the DC component. The formula for calculating ErrM is reported in (18).

$$ErrM = mean\left(\frac{\left|\hat{x}_{meas\,i} - \hat{x}_{ref\,i}\right|}{Rng_{ref}}\right) * 100; \tag{18}$$

As in (17), the error in this case is also computed considering two series of n samples, where each sample is indicated by a subscript i=1,...,n. A sample of the measured values for which the error is evaluated is indicated as $\hat{x}_{meas\,i}$ while a sample of the reference values is indicated as $\hat{x}_{ref\,i}$.

Finally A_{ref} is the normalization reference value used in the denominator of the error formula in order to compute the percentage for which the factor 100 is introduced.

For the case of the MMC model validation presented here, $\hat{x}_{meas\,i}$is a sample of the Csim capacitor voltage minus its average computed over the entire timespan of the simulation, while $\hat{x}_{ref\,i}$, is a sample of the HW MMC capacitor voltage minus its average value computed over the same time span of the Csim.

The denominator A_{ref} is the maximum peak–peak amplitude of the HW capacitor voltage.

The error $ErrM$ allows better appreciation of the differences in amplitude, which are at least three orders of magnitude below the waveform mean value.

The value of ErrM for the MMC model validation is 7.3%.

For 620 V DC bus voltage, the average value of capacitor voltage is -as expected- 103.33V.

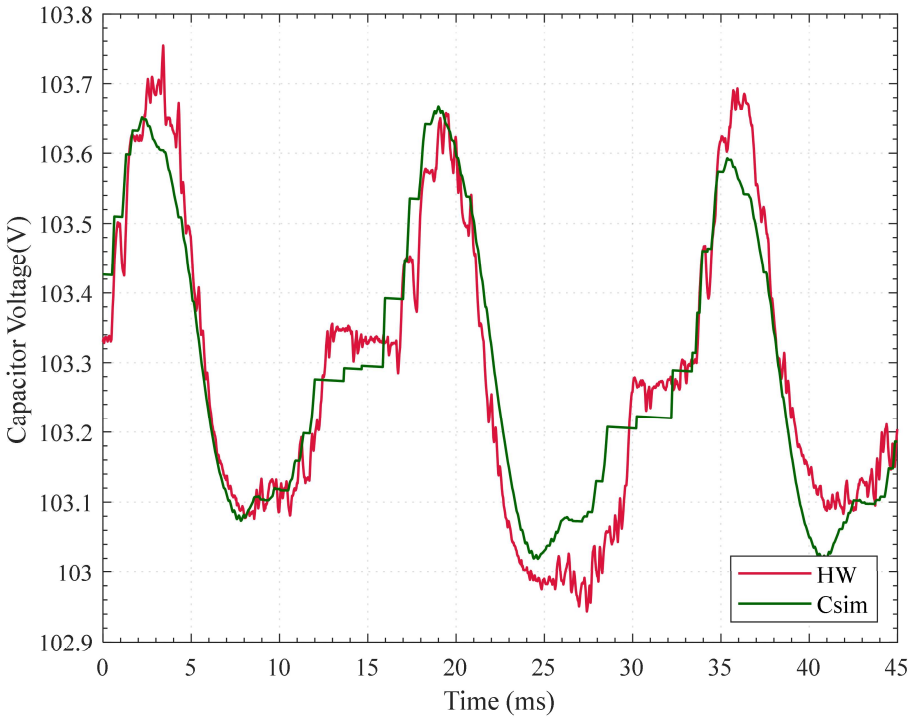


Figure 2.10 Phase A 1st capacitor voltage waveforms, Csim vs. MMC hardware prototype.

2.2.4 Scalability

Since the goal of this work is to support power electronic system level analysis while maintaining a detailed temporal resolution, it is important that we test the scalability of the proposed model. For this purpose, we considered the Xilinx Virtex-7 485t VC707 FPGA and a fixed simulation time step of 50 ns.

First, we consider the scalability of the simulator when the number of modules in a single MMC is increased.

In Figure 2.11 we report the results of the scalability analysis for a single converter when the number of levels is increased from 7 to 111.

As expected, and as highlighted by previous work on the LB-LMC method, the computational delay remains relatively constant. The variations are due to the effort of Vivado tool to best place and route the design, and the resource usage scales linearly.

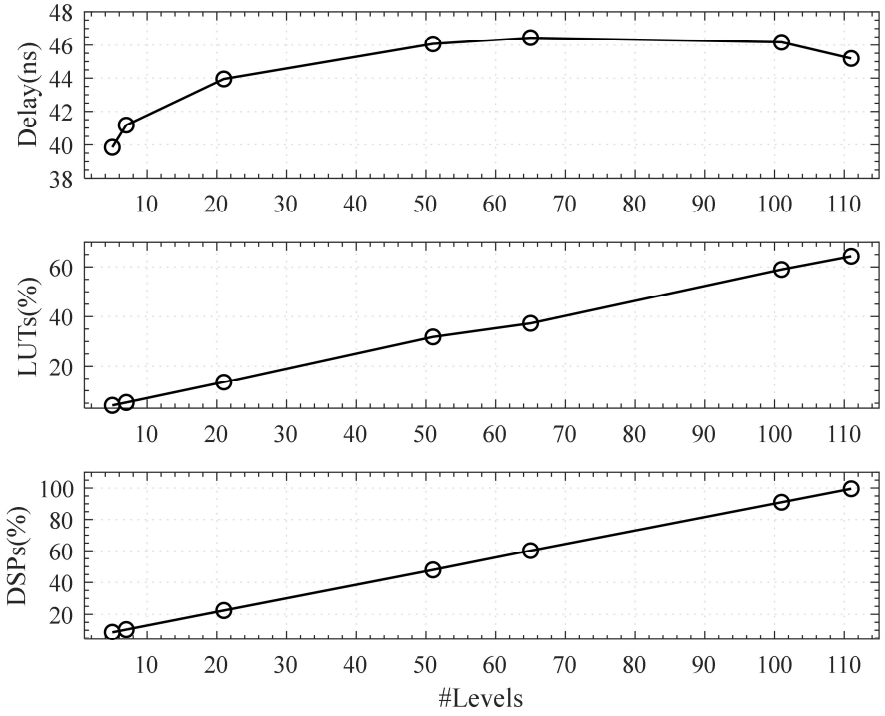


Figure 2.11 Results of the LB-LMC method scalability analysis with increasing number of MMC levels.

We also analyzed the scalability of the proposed approach when multiple converters are modelled. For this purpose, we used the microgrid of Figure 2.12, where converters are added one after the other to increase the system size.

Since we used a single cycle data flow approach to execution and due to the intuitive consideration, that similar model components require the same number of resources, we expected to observe again a linear scaling of the resource usage. Figure 2.13 shows the scalability results when an increasing number of 7-level MMC converters are added to the system.

As expected, the results show a linear relation between the system size and the resources usage.

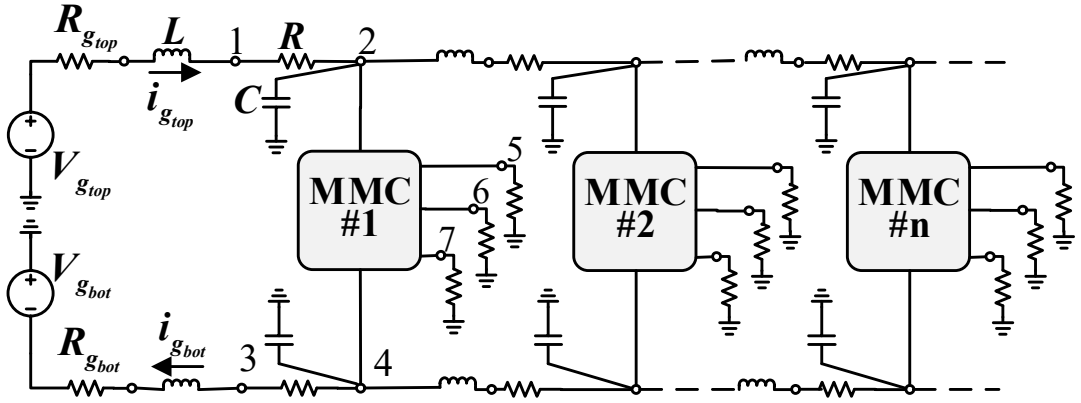


Figure 2.12 The micro grid used for the scalability analysis.

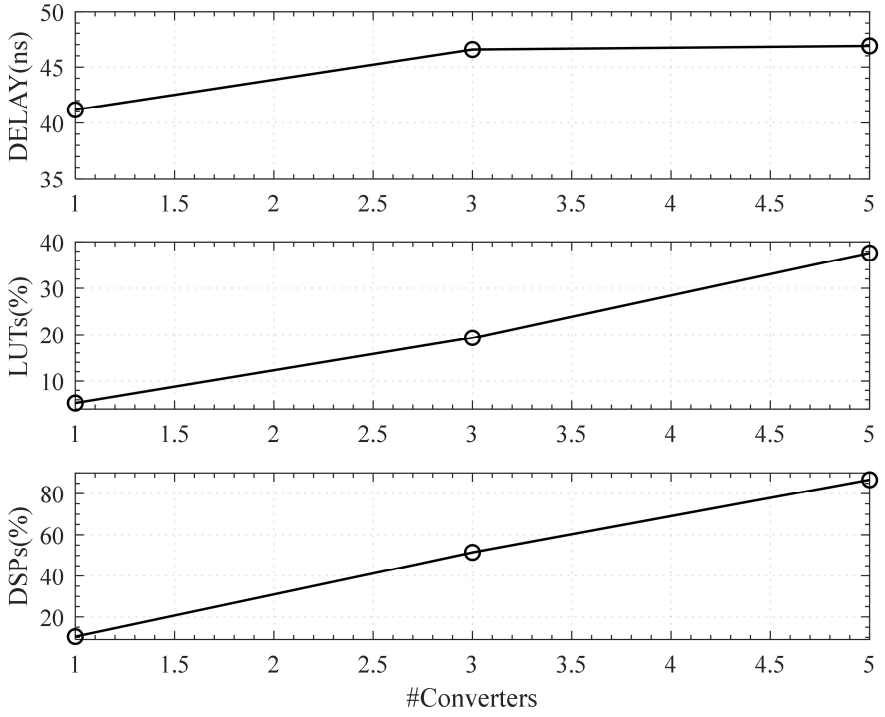


Figure 2.13 Results of the LB-LMC method Scalability analysis on Microgrids with increasing number of 7-levels MMCs.

The maximum number of switching devices present in the grid for the two-scalability analysis is much greater in the converter-size scalability analysis (660 sub-

modules) than in the case of the grid-size one (180 sub-modules). This is because in the latter case, systems with more converters have larger conductance matrixes (each converter introduces 7 more nodes) which requires larger FPGA resources allocation.

2.3 Dual Active Bridge

Similarly, to the MMC also the DAB has been found to be an interesting technology for various energy conversions applications.

Its most noticeable characteristics are the reduced use of passive elements with respect to other types of converters, the flexibility in energy transfer which makes this converter suitable to work in DC/DC, AC/AC or DC/AC and the isolation capabilities due to its internal transformer which induces a natural decoupling in the converter topology.

The DAB converter requires a control scheme simpler than the one used for the MMC since it does not require any voltage balancing algorithm with sorting of capacitor voltages and circulating current suppression, but it relies on a PSC which realizes the energy transfer from one side to the other of the converter based on phase difference in the switching actions of the two full bridge portions of the converter.

In the following sections it is described for such converter a RT simulation realized on US+ FPGA based custom platform for which a DAB full switching state space model has been implemented.

2.3.5 DAB definition and layout

A general schematic of the DAB converter considered for this work and meant for DCDC naval applications is shown in Figure 2.14. The converter combines two single phase full bridge voltage source converters (VSC) connected by a single-phase high frequency transformer and two filter capacitors at each of the two ports.

The energy transfer across the converter is cause by a modulation phase shift between the two converters sides.

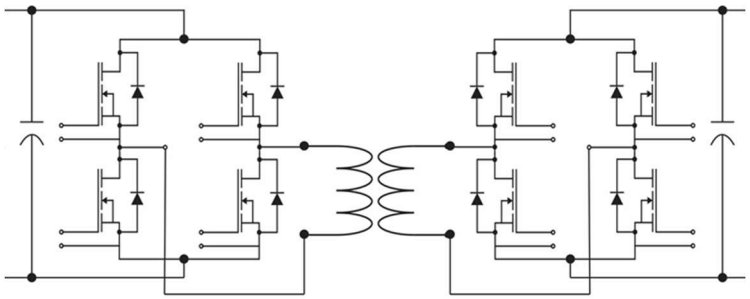


Figure 2.14 The DAB circuit diagram.

2.3.6 Literature review of some DAB models

The DAB topology modelling has attracted a considerable interest from academia and industry in the recent years which is confirmed by the considerable amount of works found in literature.

The majority of which regards general averaged models [34][35][36][37] . As per definition averaging implies modifying the real system behavior by replacing the original information provided by the switching behavior of the converter with an averaged information (i.e., the averaging of the ripple) and this procedure founds applications in control design applying the small-signal hypothesis which adds further approximations. In this category there also works for which the typical state space modelling technique has been applied like in [38] and those which propose large signal averaged models as in [39].

In [40] it is proposed a full discrete time model. Its main idea is to derive an iterative function that expresses the state variables at one sampling time by using those at an earlier sampling time.

This offers more complete information of both slow and fast dynamics of the converter and higher accuracy than the average models.

Either the averaging or the full discrete time modelling technique present high complexity from the analytical point of view. Their equations do not present a direct correspondence to the switching behavior of the real system, but this must be translated from duty cycle-based functions rather than having simple switching functions with Boolean values (0=off, 1=on). Their analytical forms are not suitable for practical implementations on high performance parallel computing-based simulation platforms like the ones based on FPGAs.

In [41] the switching states of a DAB converter are analyzed through a switching model realized using the pre-existing components of the PowerSym library of MATLAB Simulink which is as per definition a black box model which C++ code auto generated by the tool is not suitable for direct FPGA implementation.

So far no one has proposed a full switching model for the Dual Active Bridge, based on switching functions and realized with a state space modelling approach. This new model is proposed in this work and describe in the next paragraphs. The model has been implemented in C++ to be easily deployed on FPGA based simulator previous translation in VHDL using the capabilities of the Xilinx VIVADO toolchain.

2.3.7 State Space model

In Figure 2.15 it is presented the topology of the circuit used to test the developed DAB converter model in DC-DC configuration.

The circuit model is meant for RT simulation applying the LB-LMC method. Hence the nodal voltages N_1 and N_2 and the common ground 0 -for all the circuit components- are illustrated. The two DAB's bridges Bridge-1 and Bridge-2 are respectively connected to N_1 and N_2 where the voltage source V_g - R_g and the battery V_o - R_o are connected as well.

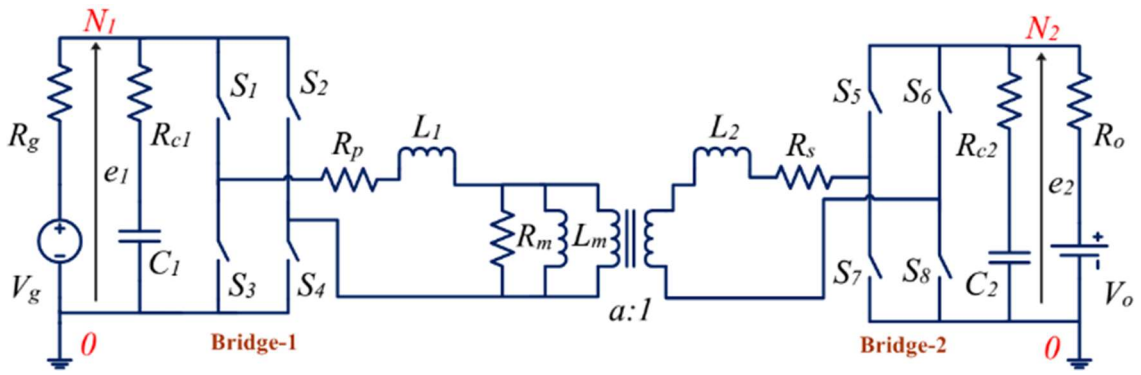


Figure 2.15 The topology of the DAB DC-DC converter circuit.

The converter architecture is symmetrical with each side of the internal transformer connecter to the respective full bridge VSC. A total of eight switching devices are present and for sake of simplicity diodes and snubber have not been implemented in the model.

The converter model has eight SPST switches in the two H-bridge sections bridge-1: S_1-S_4 and bridge-2: S_5-S_8, two filter capacitors C_1 and C_2 at the input and output end of the converter with their respective equivalent series resistors (ESRs) R_{c1} and R_{c2}, a winding resistor R_p, a leakage inductor L_1, a magnetizing core inductor L_m and a magnetizing core resistor R_m that represents the transformer core loss. There are a winding resistor R_s and a leakage inductor L_2 at the secondary side of the transformer that has a turns ratio of a:1.

The converter is fed by a real DC voltage source of value V_g with a very small source resistor R_g -simulating the non-idealities of the source- connected at node N_1.

While at node N_2 an active load V_o with a negligible series resistor R_o is connected. Since the converter will be integrated in a LBLMC network relying on nodal voltage solutions, also the nodal voltages are represented as e_1 and e_2, respectively for node N_1 and N_2 .

Energy is transferred from the input to the output ports of the converter passing through its transformer thanks to an open loop PSC control – based on PSM technique-

inducing a phase shift ϕ with a fixed duty ratio k between the switching configurations of the two H-bridge sections of the converter.

Basically, the switching signals for the two bridges will follow two shifted square waves with the one of the input bridges leading.

Considering for this model, a transformer with no losses, the delivered power can be mathematically expressed as in equation (19).

$$P_d = \frac{V_g V_o a d(1-d)}{2 f_{sw} L_{eq}} \tag{19}$$

Where d is the phase shift ratio ($\frac{\phi}{\pi}$), f_{sw} is the converter switching frequency, and L_{eq} is the equivalent leakage inductance which is equal to $L_1+a^2L_2$.

The DAB portion of the circuit in Figure 2.15 has been extracted and represented with its equivalent circuit, as shown in Figure 2.16 to be modelled with the state space method based on switching function modelling the conducting devices behavior.

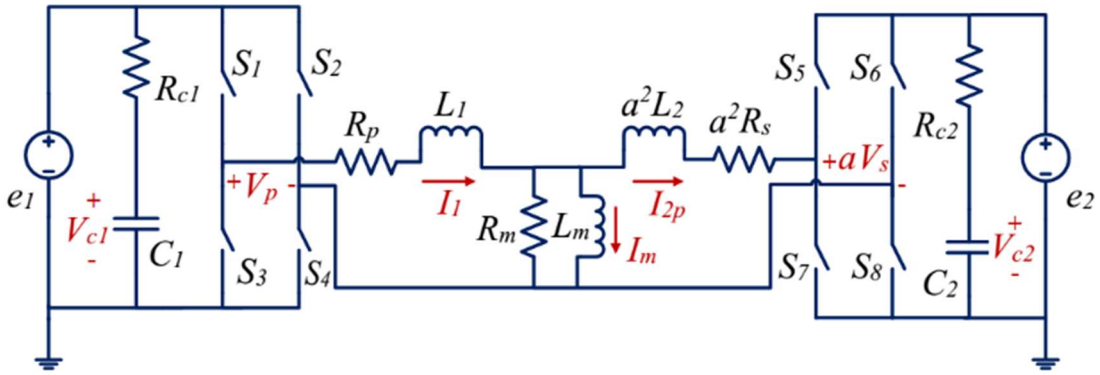
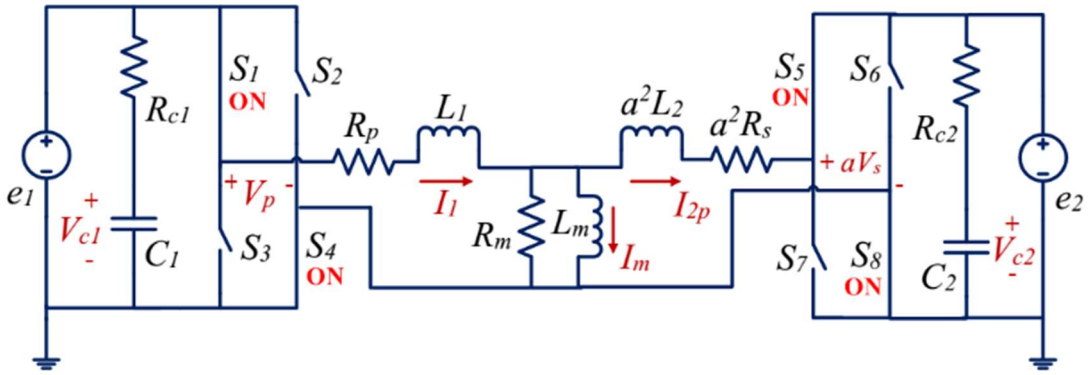


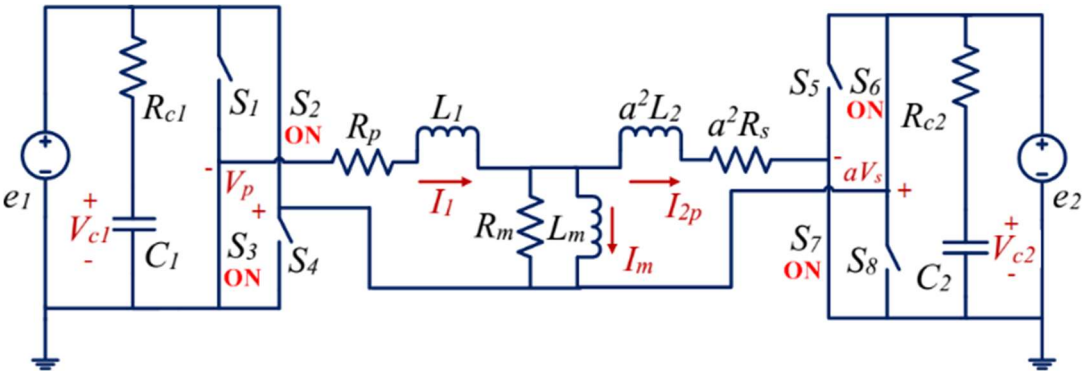
Figure 2.16 DAB converter equivalent state space model.

To make the model equations agnostic of external components characteristics, and to model the DAB terminals at which the DAB component is connected in the LB-LMC network, nodal voltages e_1 and e_2 from the remaining portion of the systems are represented with voltage sources.

The transformer is represented with its T-equivalent model and the secondary side components parameters are transported to the primary side by multiplying them with opportune coefficients. In this configuration the current flowing through a^2L_2 is $I_{2p} = \frac{I_2}{a}$. The transformer primary and secondary side voltages V_p and V_s where the latter is replaced with its referred-to-the-primary side equivalence aV_s. The model is based on five state variables, considering two capacitors and three inductors. Only the power transfer switching state configurations are considered for analyzing converter operations . Consequently, the two switching configuration are defined as $S = S_1S_4S_5S_8$ and $T = -(S_2S_3S_6S_7)$.The converter topology in configuration S is illustrated in Figure 2.17(a) .



(a)



(b)

Figure 2.17 DAB state space model in S (a) and T (b) switching configurations.

In this configuration the bridge-1 and bridge-2 voltages become - $V_p=(S_1S_4)e_1$ and $aV_s=(S_5S_8)e_2$, respectively and the state variables are derived from circuit analysis in equations (20) to (24).

$$\frac{dV_{c1}}{dt}=\frac{1}{R_{c1}C_1}(e_1-V_{c1}) \tag{20}$$

$$\frac{dV_{c2}}{dt}=\frac{1}{R_{c2}C_2}(e_2-V_{c2}) \tag{21}$$

$$\frac{dI_1}{dt}=\frac{1}{L1}[(S_1S_4)e_1-\left(R_p+R_m\right)I_1+R_mI_{2p}+R_mI_m] \tag{22}$$

$$\frac{dI_{2p}}{dt}=\frac{1}{a^2L_2}[R_mI_1-(R_m+a^2Rs)I_{2p}-R_mI_m-(S_5S_8)e_2] \tag{23}$$

$$\frac{dI_m}{dt}=\frac{R_m}{L_m}(I_1-I_{2p}-I_m) \tag{24}$$

The converter topology in configuration T is illustrated in Figure 2.17(b) where he bridge-1 and bridge-2 voltages are given by - $V_p=-(S_2S_3)e_1$ and $aV_s=-(S_6S_7)e_2$, respectively. Since the equations of the states – V_{c1}, V_{c2} and I_m derived during the *S* mode are the same, they are not repeated here.

The state variables I_1 and I_2 can be obtained from equations (25) and (26) where the latter reports the state variable I_2 reflected to the transformer primary side and named as I_{2p} .

$$\frac{dI_1}{dt}=\frac{1}{L_1}[-(S_2S_3)e_1-\left(R_p+R_m\right)I_1+R_mI_{2p}+R_mI_m] \tag{25}$$

$$\frac{dI_{2p}}{dt}=\frac{1}{a^2L_2}\left[R_mI_1-(R_m+a^2Rs)I_{2p}-R_mI_m+(S_6S_7)e_2\right] \tag{26}$$

The full switching state space model equation is obtained by combining the two set of equations inherent of each switching configuration (i.e. ,S and T) leading to (27).

$$\frac{d}{dt}\begin{bmatrix} V_{c1} \\ V_{c2} \\ I_1 \\ I_{2p} \\ I_m \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-2}{R_{c1}C_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{-2}{R_{c2}C_2} & 0 & 0 & 0 \\ 0 & 0 & \frac{-2(R_p + R_m)}{L_1} & \frac{2R_m}{L_1} & \frac{2R_m}{L_1} \\ 0 & 0 & \frac{2R_m}{a^2 L_2} & \frac{-2(R_m + a^2 R_s)}{a^2 L_2} & \frac{-2R_m}{a^2 L_2} \\ 0 & 0 & \frac{2R_m}{L_m} & \frac{-2R_m}{L_m} & \frac{-2R_m}{L_m} \end{bmatrix} \begin{bmatrix} V_{c1} \\ V_{c2} \\ I_1 \\ I_{2p} \\ I_m \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{2}{R_{c1}C_1} & 0 \\ 0 & \frac{2}{R_{c2}C_2} \\ \frac{S_1 S_4 - S_2 S_3}{L_1} & 0 \\ 0 & \frac{S_6 S_7 - S_5 S_8}{a^2 L_2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{27}$$

2.3.8 LBLMC implementation

In order to test the operation of the newly developed DAB state space full switching model of section 2.3.7, the circuit illustrated in Figure 2.15 has been simulated in RT employing the LBLMC method according which the circuit equivalent model is the one appearing in Figure 2.18 . Specifically, in such network the resistive companion equivalent of the DAB model, corresponds to two voltages sources V_{ss1} and V_{ss2} each one in series with equivalent resistances R_{ss1} and R_{ss2}.

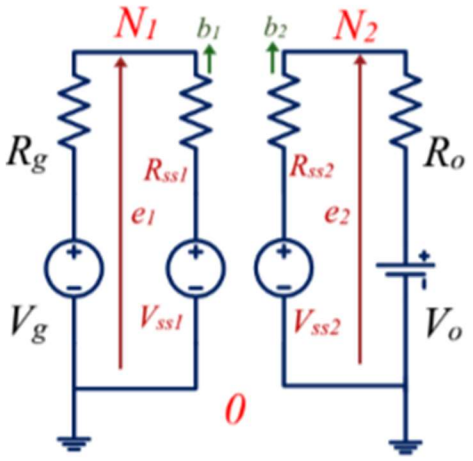


Figure 2.18 The LB-LMC network for the DAB.

Such voltage sources are updated by the nodal voltage solutions e_1 and e_2 - generated by the LBLMC solver at each simulation time step- and produce currents contributions b_1 and b_2 to their respective node.

The system conductance matrix G is represented in (28). The matrix size is 2×2 corresponding to the two independent nodal solutions at nodes N_1 and N_2 . No components are interconnected between such nodes so that G has non zero components only along its main diagonal. The vector of currents contributions is shown in (29) with size 2×1.

$$G = \begin{bmatrix} \frac{1}{R_g} + \frac{1}{R_{ss}} & 0 \\ 0 & \frac{1}{R_{ss2}} + \frac{1}{R_o} \end{bmatrix} \tag{28}$$

$$b = \begin{bmatrix} \frac{V_g}{R_g} + b_1 \\ \frac{V_o}{R_o} + b_2 \end{bmatrix} \tag{29}$$

2.3.9 Laboratory setup

To verify that the developed LB-LMC network integrating the DAB detailed model can run in RT with small time step equal to 35ns, a RT simulation experiment has been carried on employing an US+ FPGA which main features are listed in Table 2.2.

The hardware set up of the RT-simulation platform developed is shown in Figure 2.19, where the aforementioned US+ FPGA is connected to a TI-DAC unit for RT log through a Xilinx FMC adapter card plugged on connector J34 of the US+ and on the other side to the TI-DAC. On the other FMC connector of the US+ is plugged an FM-S14 card to allow fiber optic connection based on Aurora protocol for possible connections with external devices.

Table 2.2 Features of the FPGA used to simulate the DAB.

Description	Value
Brand	Xilinx
Kit board name	Virtex UltraScale+ VCU118 Evaluation Platform
FPGA's ID	xcvu9p-flga2104-2L-e
LUTs number	1,182,240
RAMs number	36.1Mb
System clock	300.0MHz

Using the VIVADO toolchain, the DAB detailed model C++ code implementation has been synthesized in VHDL code and then integrated in a VHDL test bench platform implemented manually containing all the necessary features to perform the RT simulation on FPGA. This design layout is shown in Figure 2.20.

All the main entities which are the simulation engine, the TI-DAC interface and the PSC control are implemented in a main VHDL entity representing the US+ FPGA design where instantiated are all the necessary I/O and clock ports which allows the FPGA board to communicate with the rest of the test bench.

The ports involved are the pins of the FMC connectors of which one connects the board to a Texas Instrument DAC (TI-DAC) and the other can potentially connect the FPGA with external devices such other simulators through fiber optics SFP+ connections.

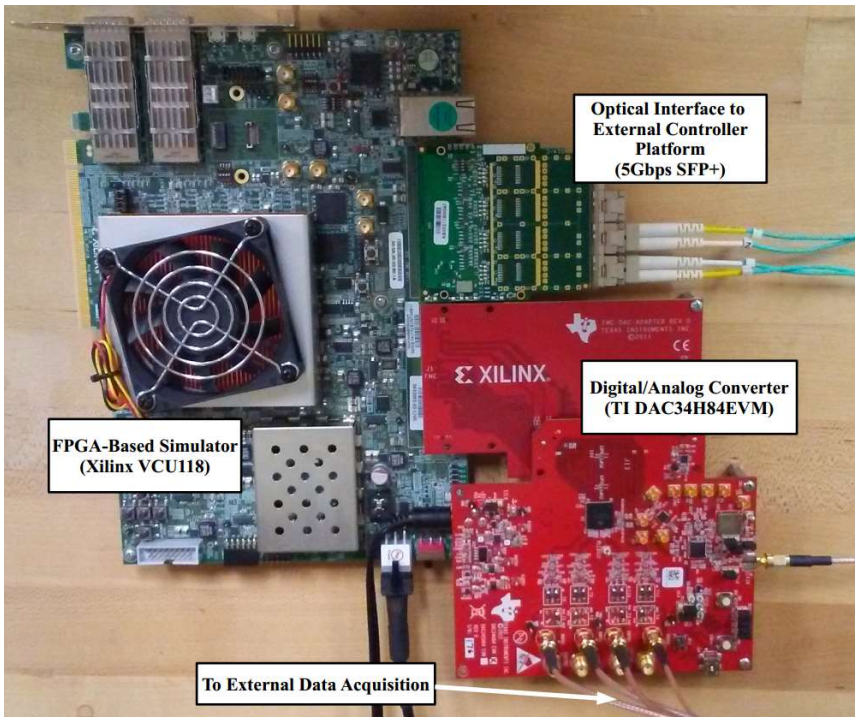


Figure 2.19 The RT Simulator Test Bench based on US+.

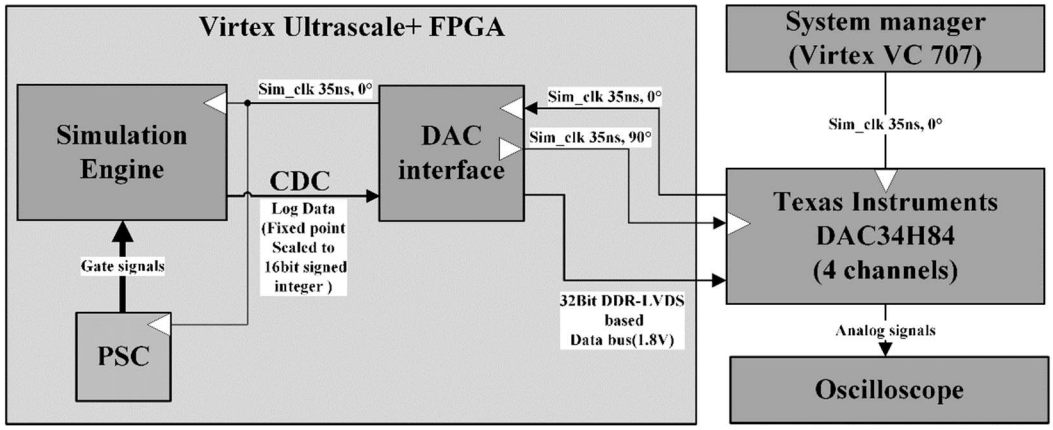


Figure 2.20 Block diagram of the simulator test bench.

In the TI-DAC interface, DDR-LVDS output converters are used to convert four 16-bit samples from the simulation engine (normalized from the engine's fixed point) into

an interleaved 32-bit DDR-LVDS data bus that is provided to the TI-DAC module to generate analog signals from the samples. The output DDR-LVDS converters are driven by a clock sourced by the DAC module which expects data to be acknowledged with a data clock 90 degrees out of phase with its own clock, so a clock generation (MMCM/PLL) block is used to create the phase shifted clock. Since the simulation engine and TI-DAC module can be driven by independent clocks which can differ in frequency and phase, asynchronous dual-clock, FIFO buffers are used to pass data without loss between the different clock domains. If the simulation engine and analog module are driven by the same clock, the FIFO buffers can be forgone, such as done in our HIL platform which uses the clock of the DAC to drive the simulation engine. Since the simulation engine may need access to digital signals, such as external switch control signals for simulated power converters, general purpose input/output (GPIO) interfaces on the FPGA are exposed to the top-level design and passed to the engine. Differently from other Xilinx FPGAs such as Kintex KC705 or Virtex 7 the Vivado library does not include a complete DDR IP core library so it was necessary to build such DDR entities using OBUFDS and OSERDES primitive and connect them with appropriate glue logic in order to meet all the required functionality and timing constraint requirements of the DAC design. The simulation engine and the PSC control module are VHDL entities automatically generated by the Vivado HLS (High Level Synthesis) tool from the C++ code implementation of the DAB detailed model. Inputs and outputs of each entity are held constants over a simulation time step through registers which are automatically placed by the synthesizer due to the particular style adopted in the C++ code. Specific HLS directives have been introduced in such a code in order for the final FPGA design to meet the requirements of zero latency and

initiation interval at most equal to one. Other directives are meant to unroll for-loops, to locate the execution of the VHDL in a specific region of the FPGA, to partition and allocate memory for arrays and to indicate to the RTL synthesizer the interfaces signals.

2.3.10 Experimental results

In this section are presented the experimental results from the FPGA RT execution of the DAB model described previously which electrical and simulation parameters are reported in Table 2.3. Such results are validated towards a MATLAB Simulink model of the same DAB converter network and then superimposed each other to appreciate the model accuracy.

Table 2.3 The DAB LB-LMC network parameters used for FPGA simulation.

Symbol	Description	Value
V_g	DC voltage source generator value	$12kV$
V_0	Active load voltage	$1kV$
R_g	DC voltage source generator resistance value	0.1Ω
R_{c1}	ESR of capacitor C_1	0.8Ω
R_{c2}	ESR of capacitor C_2	0.8Ω
R_o	Load resistance	$0.2m\Omega$
R_p	DAB primary side resistance	0.6Ω
R_s	DAB secondary side resistance	0.15Ω
L_m	Transformer magnetizing inductance	750 µH
L_1	Transformer primary side inductance	7µH
L_2	Transformer secondary side inductance	5µH
C_1	DAB input capacitance	900 µF
C_2	DAB output capacitance	270µF
P_d	DAB primary to secondary side delivered power	0.5 MW
k	PSC duty ration	0.5
ϕ	PSC phase shift	60°
a	Transformer turns ratio	12
f_{sw}	DAB switching frequency	40kHz
dt	Simulation time step	35ns

In Figure 2.21 are shown the square wave voltage plots from the RT FPGA execution of the primary and secondary sides of the DAB transformer highlighting the

phase shift ϕ induced by the PSC. The results have been properly postprocessed to remove the noise from the measurement system from the FPGA to the oscilloscope.

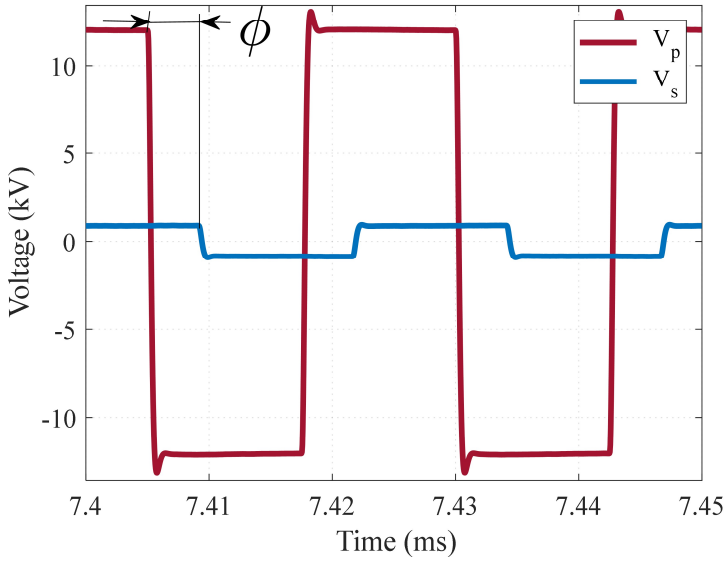


Figure 2.21 FPGA and Simulink primary and secondary side voltage waveforms showing a phase difference (ϕ = 60 degree).

The amplitudes of such waveforms are coherent with the input and output capacitor voltages of the DAB converter shown in Figure 2.22. In Figure 2.23 are shown the leakage inductors currents of the transformer primary and secondary sides and the magnetizing core induction current.

The comparison between FPGA and Simulink results leads to a two-norm error less or equal to 1% which includes the residual effect of the noise affecting the quantities logged from the FPGA using the TI-DAC and the oscilloscope.

Even though the results from the FPGA have been postprocessed to remove the noise originated by the measurement systems -where the noise has been minimized by adopting properly shielded equipment and calibrating the oscilloscope- its effect cannot totally be removed.

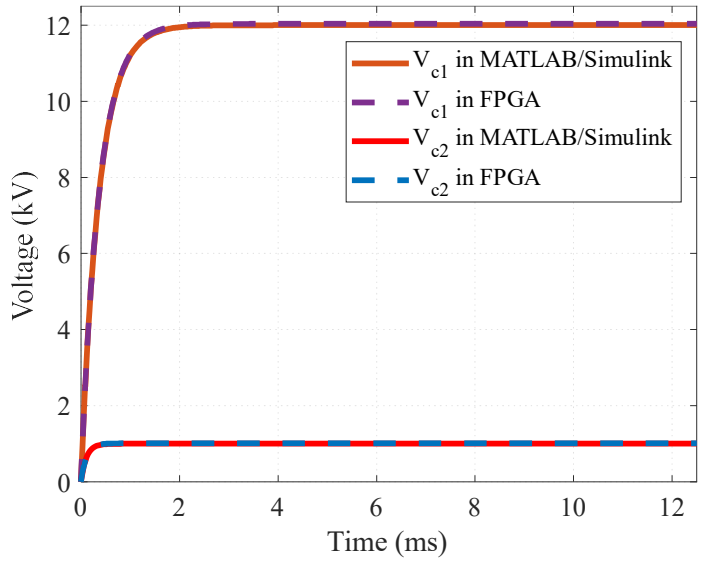


Figure 2.22 FPGA and Simulink results in terms of DAB input and output capacitor voltages.

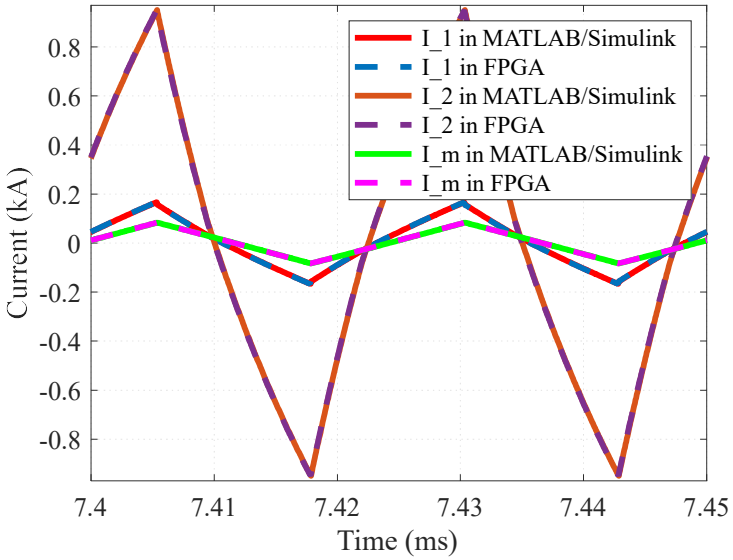


Figure 2.23 FPGA and Simulink results in terms of DAB inductors currents.

2.4 Conclusions

The MMC and DAB converters are suitable for electric ship applications where fast switching small size converters are required. In this chapter the implementation of their

models has been illustrated. The converters have been modelled applying the LB-LMC method to simulate them with small time steps specifically 50ns for the MMC and 35ns for the DAB. Both models can be simulated with high level of accuracy in a scalable and flexible way also for migrogrid with multiple converters.

CHAPTER 3

INTERFACES BETWEEN SIMULATORS

In this chapter another main contribution of this research work that is the development of communication interfaces for RT-HIL applications is described.

The serial interfaces based on Aurora protocol are meant to realize CHIL simulations and co-simulations of PEPDS.

A parallel bus interface has been developed for fast communication within the multi-FPGA platform to extend the scalability of a LB-LMC simulator of a nodal decomposed SZS model. Some application-oriented tests (i.e., RT-HIL simulations of PEPDS) have been realized to verify the operation of such interfaces.

3.1 Background

Communication interfaces between electronics devices can be either serial or parallel. Here below a brief comparison between the two interfaces according to [42] is reported. In general, referring to Figure 3.1, with a serial interface, a word of N bits is sent one bit per time through a single wire**.**

The transmission needs to be supported by a protocol to distinguish between different types of signals transmitted.

With a parallel interface the bits composing a word are simultaneously transmitted each one on a different wire, and additional bits (and consequently wires) are required to indicate validity (valid or not valid) and purpose (data, control, etc.).

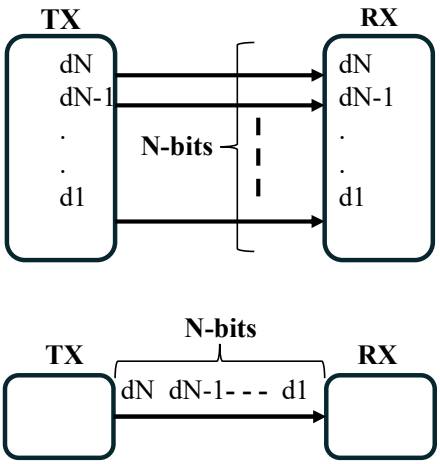


Figure 3.1 Basic parallel and serial communication interfaces schematics

Considering again a "N" bits word, a parallel interface requires N times the number of wires of the serial interface but in return it will employ 1/N the time required by the latter to transmit. The serial interface is not as efficient as the parallel one in terms of overhead since for a given word even the simplest protocol requires at least one starting and one ending bit. To send the same amount of data with same speed, the serial interface must run more than N-times the clock frequency of the parallel one.

The parallel interfaces have the disadvantage that the number of wires grows more than twice the size of the word since even for short distance to prevent disturbances (i.e., signals cross talks, noise, etc.) some precautions are needed. For example, employment of differential wiring (two wires per bit), additional wires for shielding and placement of wires at a proper distance.

The choice of the proper type of interface is affected by the desired level of communication latency, the distance, and the hardware characteristics of the communicating devices. For the present work both type of interfaces presented here are

used according to the requirements (i.e. latency, distance) of the simulation platforms involved.

3.2 Aurora 8bit interface fro CHIL testing of MMC

In this section it is presented an Aurora based interface with 8bit encoding which is meant to realize a CHIL experiment involving a multi-FPGA ring layout-based control unit based on Kintex-5 devices controlling a MMC model simulated in RT on Virtex-7 VC707 FPGA board.

3.2.1 MMC CHIL setup

The HIL set-up is composed of three HMBs in addition to the FPGA based MMC simulator with analog output capability for display of internal simulation variables. All connections between the control hardware and the MMC simulator are made via high-speed serial data channels. A picture of the CHIL set-up is shown in Figure 3.2.

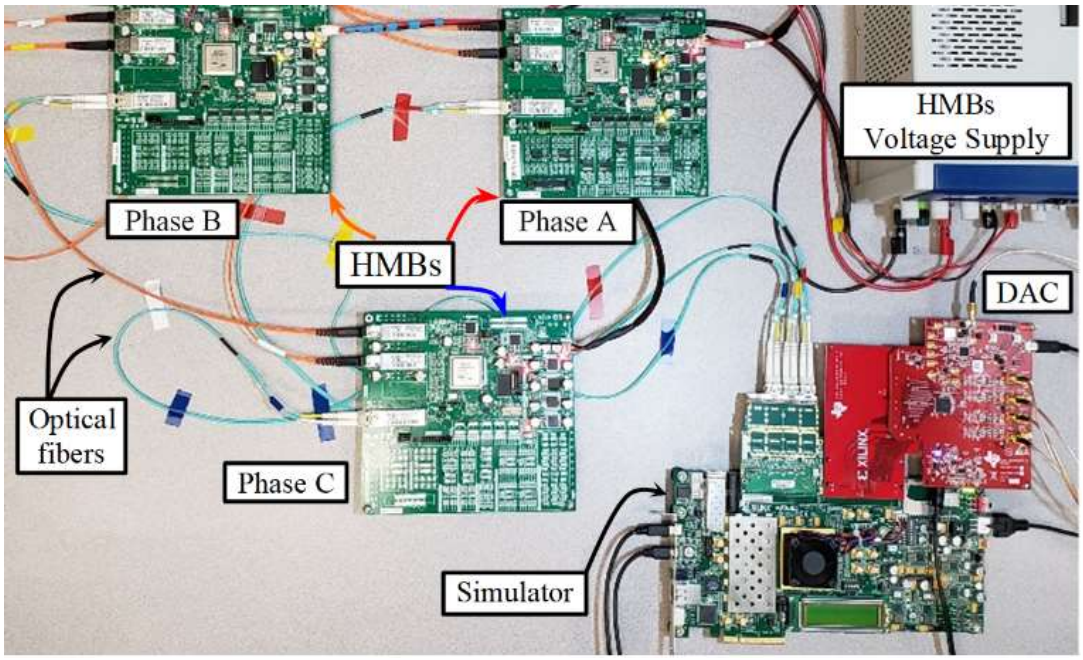


Figure 3.2Hardware-in-loop (HIL) set up using Virtex-4 Controller and Virtex-7 MMC simulator FPGAs.

A block diagram describing the simulation platform layout adopted for the CHIL experiment is reported in Figure 3.3. The system main components are the Test References generator which outputs the three phases sinusoidal reference signals {Ref}; the Phase-Shifted PWM module which generates the modulation index values {Non}; the voltage Balancing Control Algorithm (BCA) and the plant (MMC) represented by its transfer function $G_p(s)$. The section indicated by P is the section executed on the real-time simulator, the rest of the control functions, including the BCA, are executed on the control platform. The data exchanged over the fiber optics are the SM switching pulses $\{S_i\}$, the arm currents and SM capacitors voltages $\{I_{arm}, V_{cap_i}\}$.

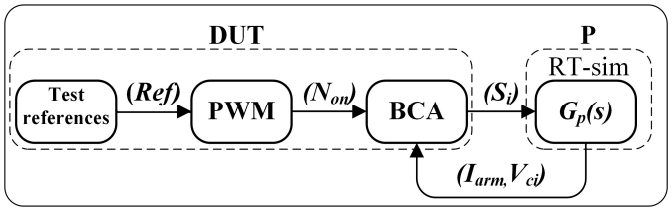


Figure 3.3 MMC CHIL experiment simulation platform layout.

3.2.1.1 RT Simulation FPGA Implementation

The MMC simulator is developed in a Virtex-7 FPGA. A FPGA top-level design was developed, depicted in Figure 3.4, that constitutes the logic of the MMC HIL simulator. The center of the design is the simulation engine, MMC Simulator, discussed in [1], which performs the real-time simulation of the modeled system. This core computes solutions of the simulated model every time step using the LB-LMC method. Using a dataflow execution approach, the simulation engine computes all solutions in a single FPGA execution clock cycle set to the time step of the simulation for real-time execution. To capture results from the simulation engine for analog output, a Digital-to-Analog Conversion (DAC) interface, based on Dual Data Rate - Low Voltage Differential Signal

(DDR-LVDS) output registers was created that captures four results (a, b, c, and d) from the engine and formats them to be streamed to an external DAC module.

The results are fixed-point 16-bit signals that produce output on the correspondent FPGA analog output port which is connected to an Oscilloscope for data capture.

The output DDR-LVDS registers are driven by a data clock sourced from the external DAC module which is used also to source the simulation engine execution clock to ensure synchronization between the simulation engine and the DAC interface/module. To allow the external controller platform access to the modeled system solved by the simulation engine, a Control Communication handler core exists in the top-level design.

3.2.1.2 Simulator Interface.

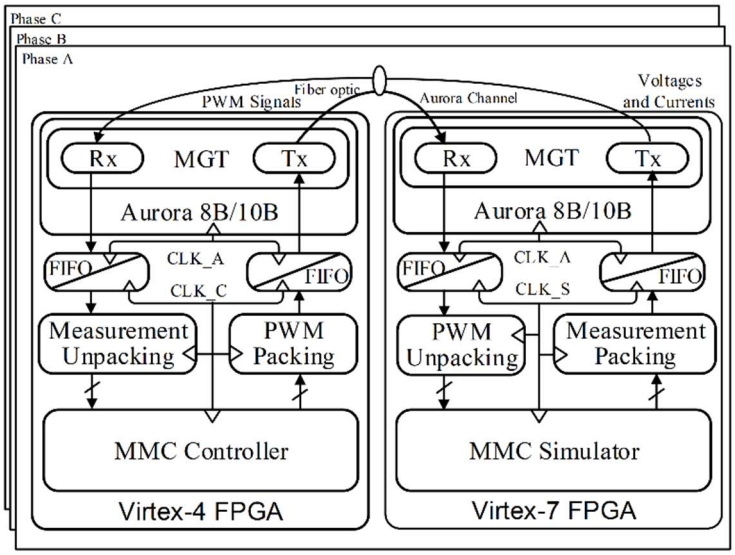


Figure 3.4 The block diagram of Aurora 8B/10B interfacing of MMC Simulator (Virtex-7) and controller (Virtex-4) per each MMC phase (A, B, C).

A high-speed communication protocol using optical fibers interfaces the simulator to the DUT. A multi-gigabit communication between the MMC simulator and each HMB utilizes the Aurora protocol [43] developed by Xilinx that uses 8B/10B data encoding.

The Aurora 8B/10B protocol has several features such as channel bonding and clock compensation that simplify the establishment of high-speed Multi-Gigabit Transceivers (MGT) based communication links.

For this work, the Aurora 8B10B protocol consists of one lane, two bytes per lane, 1.50 Gb/s line rate and 150 MHz reference clock.

In Figure 3.4 it is illustrated the block diagram of Aurora 8B/10B interfacing each of MMC Simulator phases namely Phase A, Phase B, and Phase C and the corresponding phase controller.

For CDC in-between controller or simulator and Aurora, FIFO has been used. Here, CLK_S, CLK_A, and CLK_C represents the clock for the simulator, Aurora protocol, and controller, respectively.

The Aurora system clock CLK_A has a frequency of 75MHz while simulator and controller have a frequency of 20 MHz and 125 MHz, respectively. The timing of the interface between the control boards and the simulation engine is described in Figure 3.5.

At the top of the diagram, we indicate the main entities involved. They are the controller, which generates the gates signal, the control handler, the simulation engine handler, and finally the simulation engine. Both handlers are dedicated to pack and unpack data exchanged over the Aurora interface.

After initialization of Aurora communication, each controller HMB transmits gate signals to the simulation engine.

The simulation engine receives the gate signals and sends out the measurements to the control as depicted in Figure 3.4. In Figure 3.5 the white boxes indicated with letters, represent the processes performed by each entity. Each process is described below.

Process A

This process performs the generation of PWM gate signals and then MMC capacitors voltages sorting and balancing. Process A has a time of T_0.

The HIL interfacing between phase controller and simulator happens continuously in a controlled manner.

Process B

This process takes care of packing the gates signals and of the clock domain crossing (CDC) between the controller clock CLK_C and the Aurora system clock CLK_A using FIFO as depicted in Figure 3.4.

Then the Aurora protocol transmits gate signal to the simulator FPGA platform.

Process C

This process takes care of unpacking the gates signals from Aurora channel and of the CDC between the Aurora system clock CLK_A and the simulation engine clock CLK_S.

This process triggers as soon as it receives the desired header from the controller.

Process D

This process samples the MMC capacitor voltages and arm currents measurements from the simulation engine and packs the samples.

Furthermore, it also takes care of the CDC between CLK_S and CLK_A.

Process E

This process takes care of unpacking the MMC measurement samples from simulation engine and of the CDC between CLK_A and CLK_C using FIFO as illustrated in Figure 3.4.

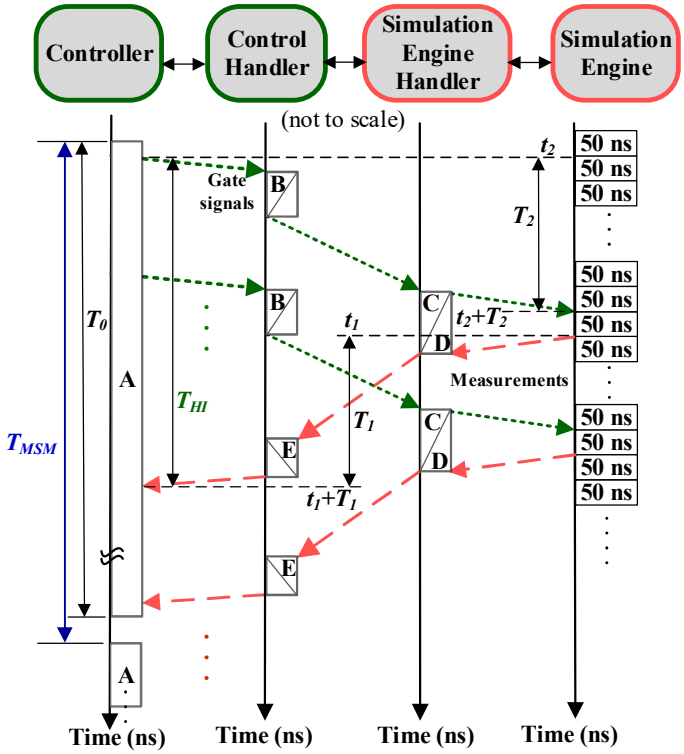


Figure 3.5 The Control-Plant timing interfacing diagram.

3.2.1.3 Communication Delay Estimation

Once the final PWM gate signals have been produced in Process A, those are sent out to Process B. Ideally, the gate signals should have been received by the simulation engine at time "t_2" but due to the delay introduced by the controller transmitting CDC and Aurora based communication system, those are received at time "t_2+T_2". The delay T_2 has been estimated to be approximately equal to 1.7µs. The delay T1 is the time required for the MMC measurements from the simulator to be available in the controller and it has been estimated to be approximately equal to 2.1µs. It is greater than T_2 since the amount of MMC measurement data to be sent out from the simulation engine is greater than the amount of data, which must be sent out by the Control (the gates signals). The HIL interfacing time T_{HI} is estimated as the summation of T_2, T_{sim}, and T_1. The magnitude of

T_{HI} is calculated at approximately equal to 4 µS. T_0 is equal to T_{MSM}-T_{HI}.The introduced delay will affect the HIL experiment accuracy. In the next section we will evaluate the impact of these delays.

3.2.2 Communication based CHIL accuracy.

In this section -so to determine the accuracy of the CHIL- the results obtained by the CHIL tests are compared with the Csim results validated in the previous section.

As we will show, the impact of the communication delay is quite small, to better appreciate the impact of the delay we decide to compare the results obtained by the CHIL experiment with the one of the Csim. In Figure 3.6 the phase A voltage waveform output from Csim and from the CHIL experiment are superimposed.

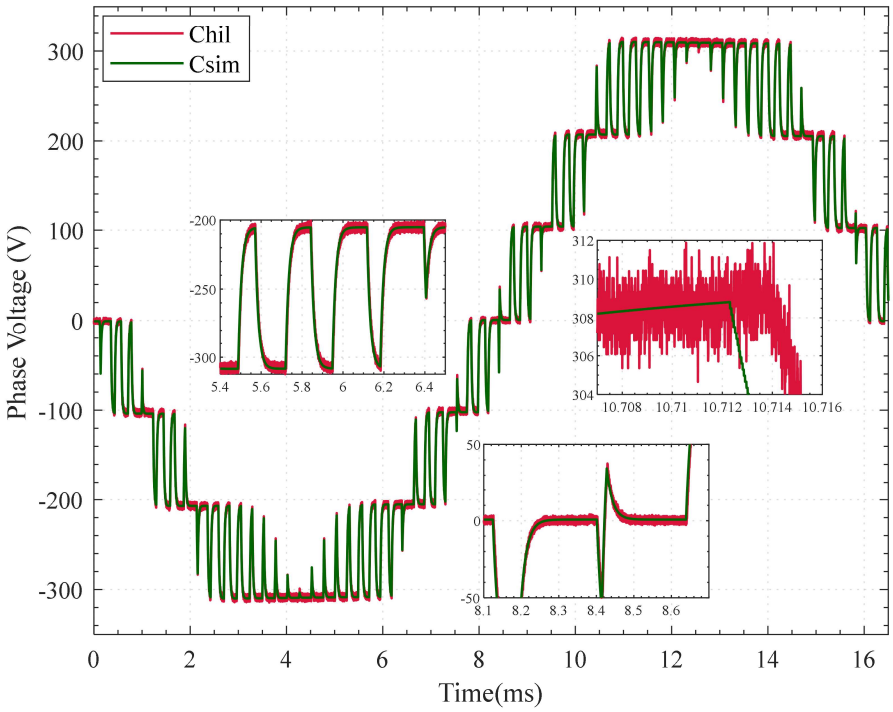


Figure 3.6 The C-simulation and CHIL voltage waveforms of phase A superimposed to highlight the effect of the CHIL on the simulation.

The main difference between the Csim and the CHIL results is that in the latter, the use of a communication protocol (Aurora) as CHIL interface, induce the communication delays T1 and T2, Figure 3.5. They cause the CHIL waveform to be delayed with respect to the Csim waveform by 2μs, as can be observed in the detailed zoom of Figure 3.6. Overall, the inaccuracy introduced by the communication delay is small and the two-norm error in this case is equal to 1.47%.

Similarly in Figure 3.7 is shown the phase A 1st capacitor voltage waveforms, in this case the error as computed according to (16) is equal to 2.87%.

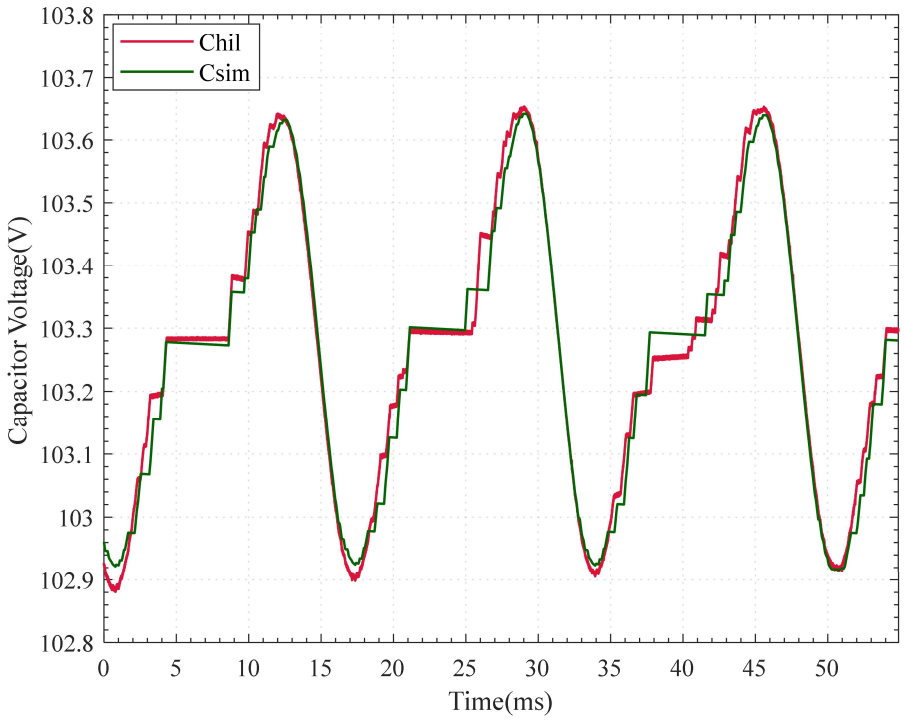


Figure 3.7 The phase A 1st capacitor voltage waveforms of the Csim and CHIL superimposed each others on.

The presence of the communication delays represents a limitation for the CHIL platform especially about the delay between when the controller updates the state of the gate signals and when the update values are received by the simulator, T2.

The negative effect of this delay will be more significant for converters switching at higher switching frequencies.

To quantify the impact of communication delay on CHIL error, a new C-simulation, namely CsimT1T2, emulating the effects of the communication delays T1 and T2, has been developed and executed for MMC with increasing switching frequency varying in a reasonable range of values in line with state-of-art conducting devices technology.

For each case the two norm error considering the Csim as reference has been computed and the results are shown in Figure 3.8 where the two norm error increases with the switching frequency within a reasonable frequency interval for SiC switching devices.

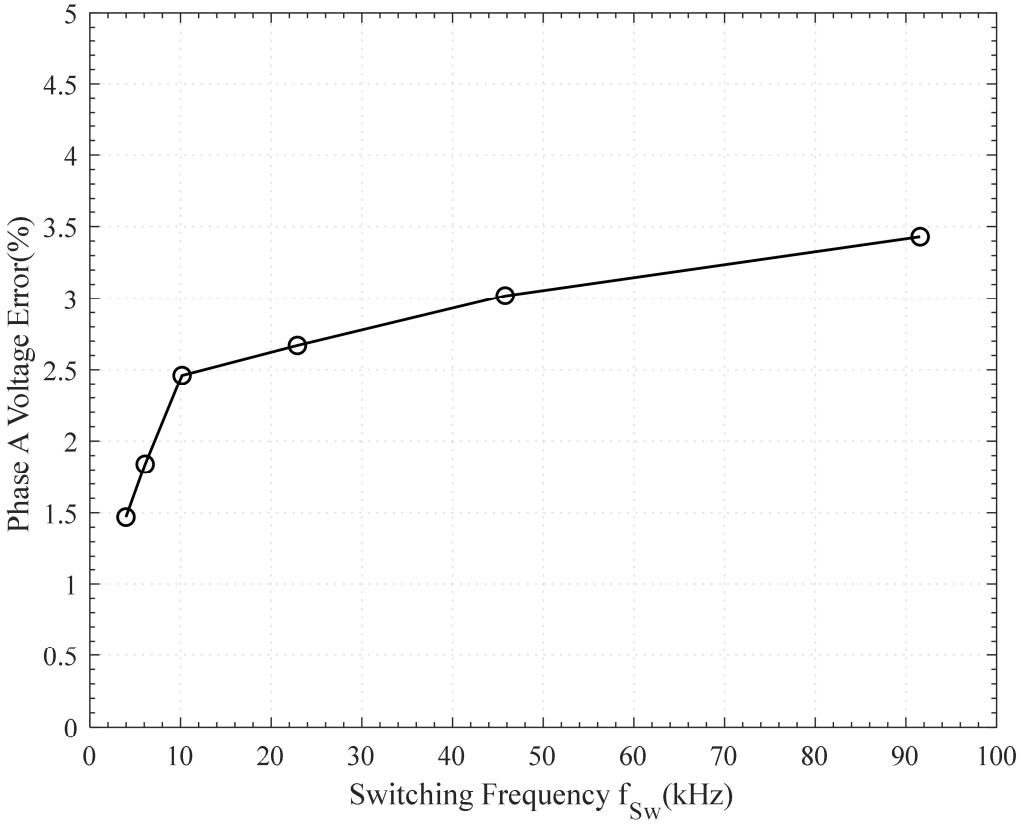


Figure 3.8 Error between the phase A voltage of the Csim and CHIL for different switching frequency f_{sw}.

3.1 Aurora 64bit interfaces for co-simulation, multi-FPGA and CHIL platforms

In this section it is described the implementation of the Aurora based interfaces with 64bit encoding adopted to realize the co-simulation method and the CHIL experiment presented in chapter 5. The implementation of this type of interface is necessary since the multi-FPGA platform is based on US+ FPGA boards implementing only the 64bit Aurora enconding.

3.1.3 Introduction

The Aurora protocol implemented by Xilinx provides a fast modular light weight serial protocol which can be employed for QSFP/SFP+ fiber optic interfacing between FPGA boards. For this work some communication interfaces based on this protocol have been implemented either to allow communication between a CPU based platform -OPAL-RT 5607- and multi-FPGA based platform to realize a new co-simulation RT-HIL method and to realize a CHIL experiment enabling the communication with an external control unit.

The communication latency of several nanoseconds characterizing those type of interfaces has negligible effect for communication between fast systems running with small time steps on FPGA and slower systems running on external devices.

Examples of those slow systems can be an external control unit for fast switching converters or an electromechanical system characterized by a slow dynamic running on a CPU based unit.

In terms of scalability this type of interface simplifies the communication between platforms since even a huge number of signals organizes in a frame can be exchanged on a single channel.

3.1.4 Architecture description

In this section it is described the internal architecture of the Aurora interface considering the basic component for general purpose based on Aurora 64bit encoding meant for the US+ FPGA and its extension on Virtex-7 -the Link Aurora.

The Aurora interface developed for this work aims to allow the US+ FPGA -hosting a fast RT-HIL simulator - to communicate with external slower dynamics systems.

The diagram of a basic building block of such interface is reported in Figure 3.9 where the typical FPGA design for RT-HIL simulations related to this work is illustrated.

Specifically, a simulation engine meant for simulations of PEPDS based on fast switching devices optimized with fixed point data format requires to interact with the rest of the simulator running on an external device and eventually also with an external control unit.

To achieve these goals said simulation engine is connected to the “Ultrascale+ Aurora Interface” design indicated in the diagram.This interface is based on the Aurora64b66b IP core provided by the toolchain Xilinx Vivado.

To proper operate, it requires a packing and unpacking “Routing Logic” illustrated in the diagram based on a RX FSM and TX FSM which technical description is provided in the next paragraphs. A CDC interface based on FIFOs synchronizes the operations of the Aurora64b66b IP core based on the FPGA board MGT transceivers clock and the simulation engine running on the simulation clock with period of 35ns.

In the case of co-simulation with external devices with a slower simulation rate a multi-rate smoothing unit is required as well as a ITM to decouple the fast simulation engine from the external slow simulator.

The data type converter module converts the IEEE 754 Single Precision Floating point data type coming from the Aurora64b66b interface into custom fixed-point format.

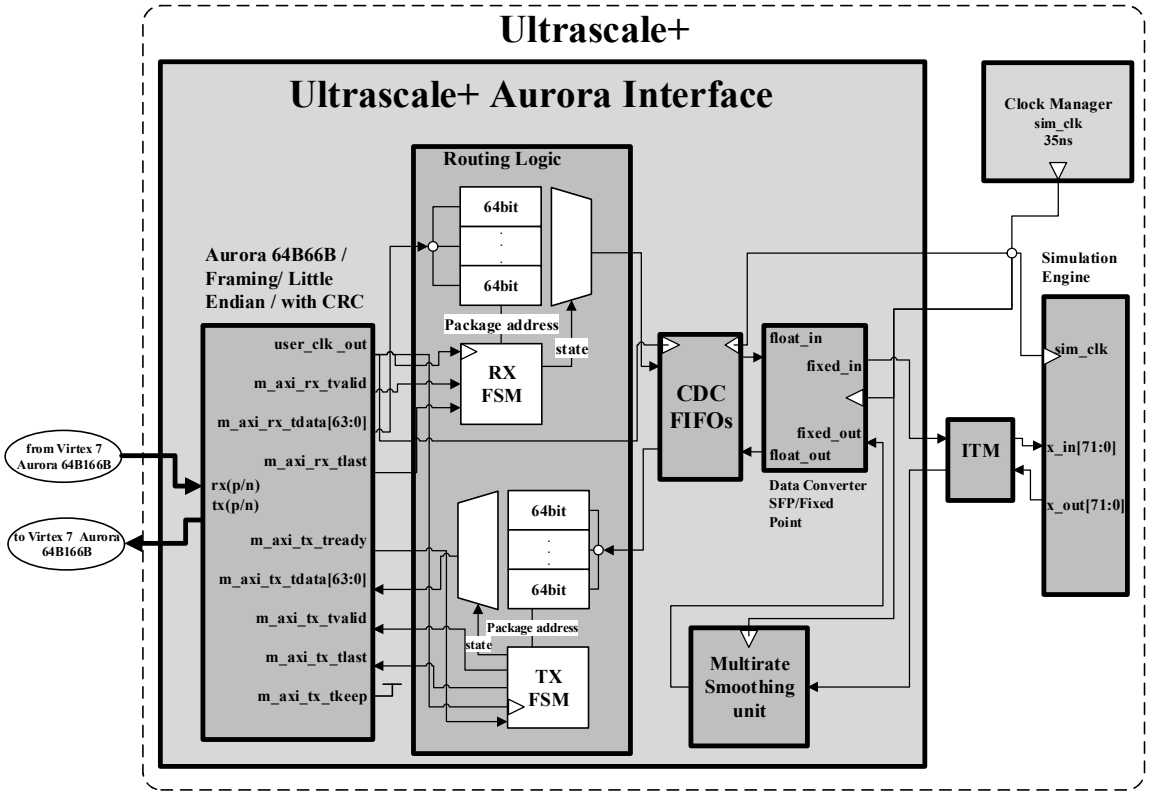


Figure 3.9 A block diagram showing the logic of the Aurora 64B66B interface on US+.

While for CHIL applications the system architecture simply requires to have this interface on both plant and control platforms, for co-simulation applications an additional logic might be required.

In the case of this work, a co-simulation platform has been realized by connecting a CPU based platform, the OPAL-RTOP5607 and a multi-FPGA platform based on US+.

For this purpose, a link design has been added to the Aurora64b66b interface to allow the communication with the hardcoded Aurora8b10b interface of Opal.

A block diagram of this additional design is illustrated in Figure 3.10. In this diagram are illustrated the respective Xilinx IP cores of the two different Aurora protocol

encodings: the core with 8b encoding communicating with Opal and the core with 64b encoding communicating with US+. In between a packing/unpacking logic handles the data words with different bit-size of the two IP cores through a CDC interface based on FIFOs.

The overall co-simulator communication interface acting as a communication link between the two types of simulation platforms is illustrated in Figure 3.11.

Besides the Aurora interface and the link logics previously described here it is illustrated how the co-simulation is realized.

The Opal platform simulated on a multi-CPU unit a slow system namely “Plant” which has a simulation time step of 25µs.

This time step is quite large in comparison with the small-time steps of the simulation engine on US+. The communication latency of the Aurora interface is below the 25µs of the Opal simulation time step. Due to these factors, the employment of serial communication and its latency are perfectly suitable for the co-simulation applications considered for this work.

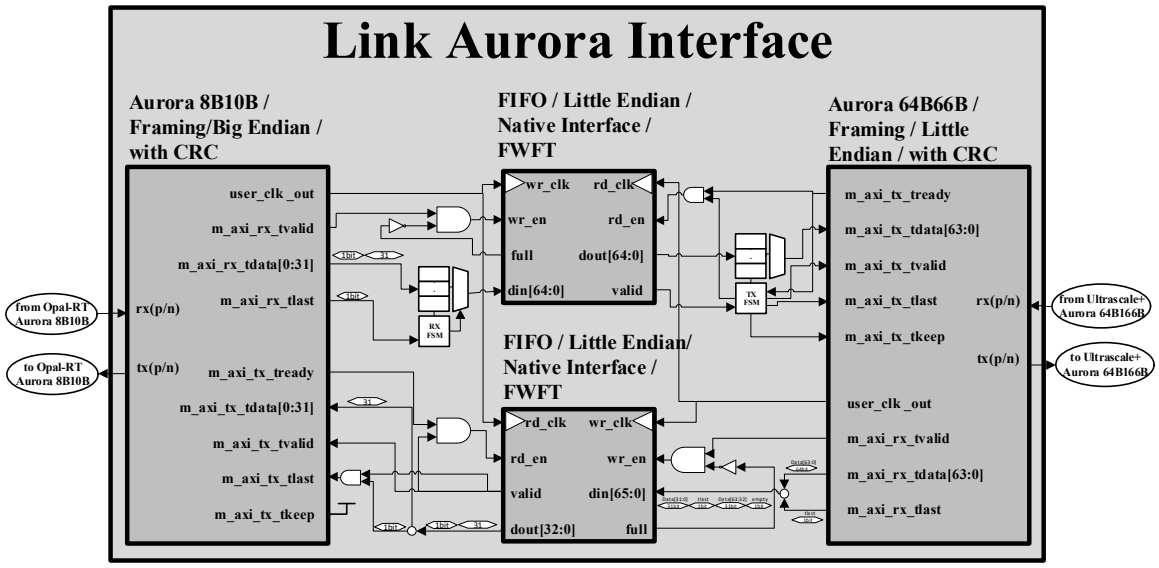


Figure 3.10 A block diagram showing the structure of the Link Aurora interface design for Virtex-7.

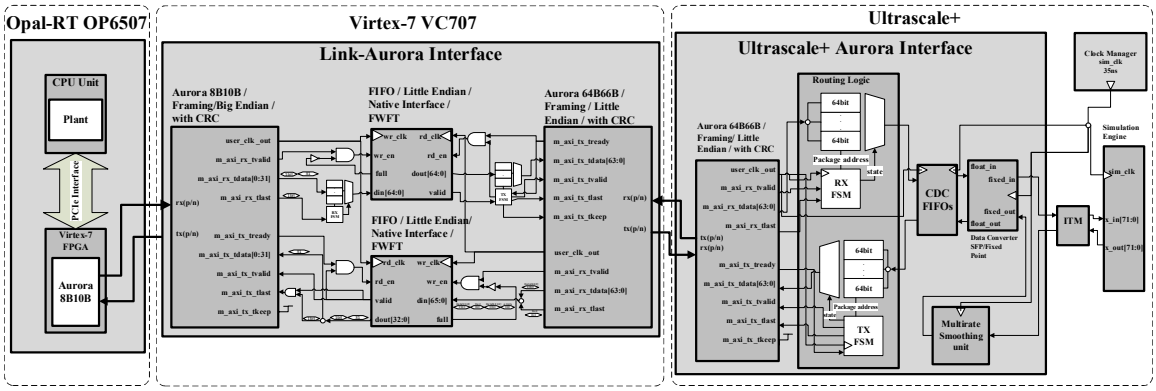


Figure 3.11 The Communication link FPGA design schematic between OPAL-RT and US+ FPGA.

3.1.5 Multi-rate smoothing interface

The co-simulation method proposed in this work involves systems with different simulation time steps where the slow system cannot gather all the information generated by the fast system.

The state-of-the-art strategy adopted to overcome this issue consists in the application of the multi-rate technique. Hence a multi-rate unit as indicated in Figure 3.9 is required to prevent loss of information during the communication.

A multi-rate simulation as nvisioned in Chapter 1, is characterized by a slow subsystem running with a simulation time step dT greater than the simulation time step dt of a faster subsystem.

This is summarized in Figure 3.12. Within a time interval dT, excluding the initial sample x_0 coming from the previous time interval, a number of samples “n” equal to (dT/ dt) is generated by the simulation engine.

Of this “n” samples, without the multi-rate function, the last sample x_n having value equal to y_n computed by the faster subsystem would have been sent to the slower subsystem but this sample is not well representative of the remaining ones.

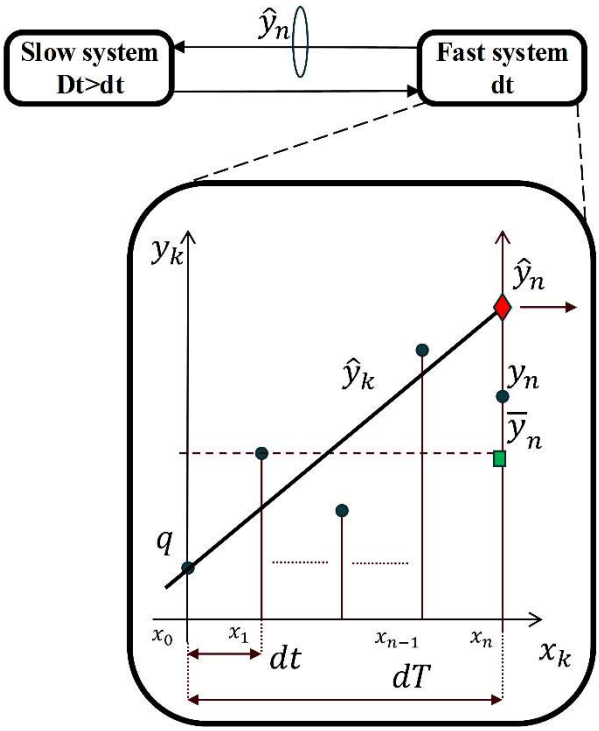


Figure 3.12 A simplified schematic of a multi-rate system with multi-rate Smoothing algorithm applied.

In order to increase the quality of this information and making this sample more representative, in other words to smooth the difference between communicating systems running at different rates, a multi-rate smoothing algorithm theoretically described in [44] has been implemented and employed for this work.

The faster subsystem generates within the time interval dT (simulation time step of the slow system) a series of samples with values y_k with k=1 to n at each point in time x_k.

The main idea of the linear multi-rate algorithm proposed here is to compute a linear approximation $\hat{y}_k$, of the series of y_k samples and use it to generate at time point x_n a value $\hat{y}_n$ which will be sent to the slower subsystem instead of the actual value y_n computed at that time by the fast system.

The linear approximated value $\hat{y}_n$, as can be seen from the picture above, will be more representative than y_n and $\bar{y}_n$ (the mean values of the samples) of the whole behavior of the fast system within the time interval dT (simulation time step of the slow system) since it follows the trend of the previous samples within dT.

In more details the linear approximation given by equation (30), requires computing the slope of the linear approximation "m" and the intercept "q".

Those values can be derived minimizing the error function generated by computing the difference between real and approximated data as done in (31) according to equation (32) to (35) where n is the number of intervals computed as $n=dT/dt$ and the summation indexing, differently from the original definition starts from 0 rather than 1 to be more close to the C++ implementation.

$$\hat{y}_k = mx_k + q \tag{30}$$

$$y_k - \hat{y}_k = y_k - mx_k - q \tag{31}$$

$$J = \sum_{k=0}^{n-1} (y_k - mx_k - q)^2 \tag{32}$$

$$\frac{\partial J}{\partial m} = 2\sum_{k=0}^{n-1} (y_k - mx_k - q)(-x_k) \tag{33}$$

$$2\sum_{k=0}^{n-1} (y_k - mx_k - q)(-x_k) = 0 \tag{34}$$

$$m = \frac{\sum_{k=0}^{n-1}(x_k y_k) + q\sum_{k=0}^{n-1}(x_k)}{\sum_{k=0}^{n-1}({x_k}^2)} \tag{35}$$

Equation (30) and (35) have been implemented in C++ code to be further synthesized in VHDL using fixed point format and in order for this module to have a zero latency and initiation interval equal to one the denominator of (35) has been pre computed

offline to eliminate the division operation from the C++ code which once synthesized in VHDL requires a multi-cycle type operation module with latency which can go up beyond 200 cycles.

For more accurate results especially for high values of n, it is also possible to adopt a more accurate and more computation cumbersome quadratic approximation reported here below in equations (36) to (44).

$$y[i] = a * i^2 + b * i + c \tag{36}$$

The coefficient a, b, c of (36) are the typical of the quadratic regression formula based on least square method , their expressions are reported in equations (37) to (39). Those coefficients are computed using the coefficients defined in equations (40) to (44).

$$a = \frac{\delta * \alpha - \beta * \gamma}{[\alpha * \varepsilon - \gamma]^2} \tag{37}$$

$$b = \frac{\beta * \varepsilon - \delta * \gamma}{\alpha * \varepsilon - \gamma^2} \tag{38}$$

$$c = \frac{\sum_{i=1}^{n} y_i}{n} - b * \frac{\sum_{i=1}^{n} i}{n} - a * \frac{\sum_{i=1}^{n} i^2}{n} \tag{39}$$

$$\alpha = \sum_{i=1}^{n} i^2 * \left(1 - \frac{1}{n}\right) \tag{40}$$

$$\beta = \sum_{i=1}^{n} y * i - \frac{1}{n} * \sum_{i=1}^{n} i * \sum_{i=1}^{n} y \tag{41}$$

$$\gamma = \sum_{i=1}^{n} i^3 - \frac{1}{n} * \sum_{i=1}^{n} i * \sum_{i=1}^{n} i^2 \tag{42}$$

$$\delta = \sum_{i=1}^{n} y * i^2 - \frac{1}{n} * \sum_{i=1}^{n} i^2 * \sum_{i=1}^{n} y \tag{43}$$

$$\varepsilon = \sum_{i=1}^{n} i^4 - \frac{1}{n} * \left(\sum_{i=1}^{n} i^2 \right)^2 \tag{44}$$

3.1.6 Aurora finite state machine description

In this section, it is reported a conceptual explanation of the Aurora TX and RX FSMs operations with corresponding schematics which purpose is to show what happens during each state of the system during rising and falling edge of the clock driving the FSMs (user clock, “user_clk”).

The type of serial communication through Aurora employed for this work requires a custom logic to handle the reception and transmission of data packet according the ready then valid hand shaking philosophy adopted by the Aurora protocol

On the transmitting side, the TX FSM, shown in Figure 3.13 works on the rising edge of the Aurora user clock and must handle cases when the Aurora core becomes not ready at some point along the frame transmission during the clock compensation routine.

When this happens, the TX FSM must be capable to retain the last transmitted value until the transmitting condition are re-established (i.e., tready and tvalid both asserted).

The state machine is automatically triggered and starts its operations as soon as the bitstream of the FPGA design is deployed on the device.The states are described here below. Besides the handshaking signals appearing into the FSM diagram there is one important parameter “*state*” which numerically indicates the frame word number and spans from 0 to 15 being the frame composed of 16 words each one of 64bit which is created by

the user logic handling a single word of 64bit or by packing two consecutive 32bit words. This depends on the needs of the external communicating systems.

OFF

The system remains in this state during reset operation when the RESET signal is asserted and exits when the reset is de-asserted.

IDLE

The system transitions to this state when the Aurora core is not ready to transmit (tready = 0), for example when the core must perform the clock compensation procedure.

The data is not valid during this time interval so the last valid data from previous transmission states is held to be transmitted whenever the system will be ready again.

INITIALIZATION

The system transitions to this state when the Aurora core is ready to transmit (tready =1) or after the OFF state or after the end of the transmission of a previous frame which is determined by the END OF FRAME state, when the state variable is set to zero. In this state all the system parameters are initialized to start transmitting a new frame.

FILL FRAME

In this state the Aurora core is ready and the frame is filled with the data words.

Each word is filled with data from user logic on the rising edge of the user clock until the end of frame is reached.

The system remains in this state when “state” parameter is less than frame size represented by the parameter FRAME_SIZE (the *state* range is [0; FRAME_SIZE-1]).

It then exits this state and reaches the END OF FRAME state when the end of the frame is reached when the tlast flag becomes equal to 1.

HOLD

The system returns in this state from every other state on the falling edge of the user clock. As per their natural behavior, signals inside the FPGA do not persist during the falling edge transition unless explicitly indicated a hold command.

END OF FRAME

The system enters in this state when the last data word has been fed to the frame so the tlast flag is asserted by the user logic and exits this state when the state parameters is set equal to zero by the user logic.

The FSM behavior is based on the “ready than valid” philosophy. The user logic waits for the Aurora core to assert the ready signal and the decide to start the communication by providing to the core the valid signal tvalid which when asserted indicates the validity of the transmitted word, the tlast and tkeep.

The user logic in this case has more control on the communication with respect to the receiving side of the system where Aurora core does not provide any ready signal but as soon as data are gathered in by the core the data are delivered to the user logic together with tlast, tvalid and tkeep signals.

A schematic of the RX FSM is shown in Figure 3.14 , where is possible to observe that the RX FSM has a logic similar to the TX FSM and works on the rising edge of the user clock with similar operational states and transitions. As in the previous case the FSM is triggered upon deployment of the design on the FPGA.

Differently from the other case there is not a ready signal coming from the master, but the data reception is triggered upon reception of a valid flag indicating that the incoming data are valid.

Whenever the received data are valid, the machine updates its internal state in which a specific packet address numbered from 0 to 15 is selected, at that point the received 64bit data packet is placed in a specific position of an array which constitutes the frame.

The TX and RX FSM operation described so far has been employed to develop various type of interfaces based on the 8bit and 64bit Aurora encoding

For each case a different Xilinx VIVADO Aurora IP core has been employed; according to the encoding Aurora8b10b IP core with a data word of 32bit or Aurora 64b66b IP core with a data word of 64bit

While the Aurora8b10b allows the selection of different word bit sizes, the 32bit size has been selected for this work to handle the floating point data type of the Opal-RT internal hardcoded Aurora interface.

Specifically, the Opal-RT test bench embeds an internal Virtex-7 FPGA where a communication interface based on Aurora8b10b is hardcoded besides this it is also hardcoded the frame size which consists of 32 words.

Such interface exchanges data words in floating point format with a bit size of 32bit compatible with the Aurora8b64b IP core while on US+ FPGA the available Aurora64b66b IP core requires a data word bit size of 64bit and being this interface custom made the frame size has been selected to contain 16 words.

Hence according to the application, the Aurora interface FSMs described in this section have been adapted to the required frame size.

In the case of the Aurora 64bit interface for co-simulation, the RX FSM is supported by a packing logic to concatenate two consecutive 32bit data words in one 64bit data word fed to the Aurora64b66b core.

While the TX FSM is supported by an unpacking logic to split a single 64bit data words in two consecutive 32bit data words fed to the FPGA design internal logic (i.e., the simulation engine).

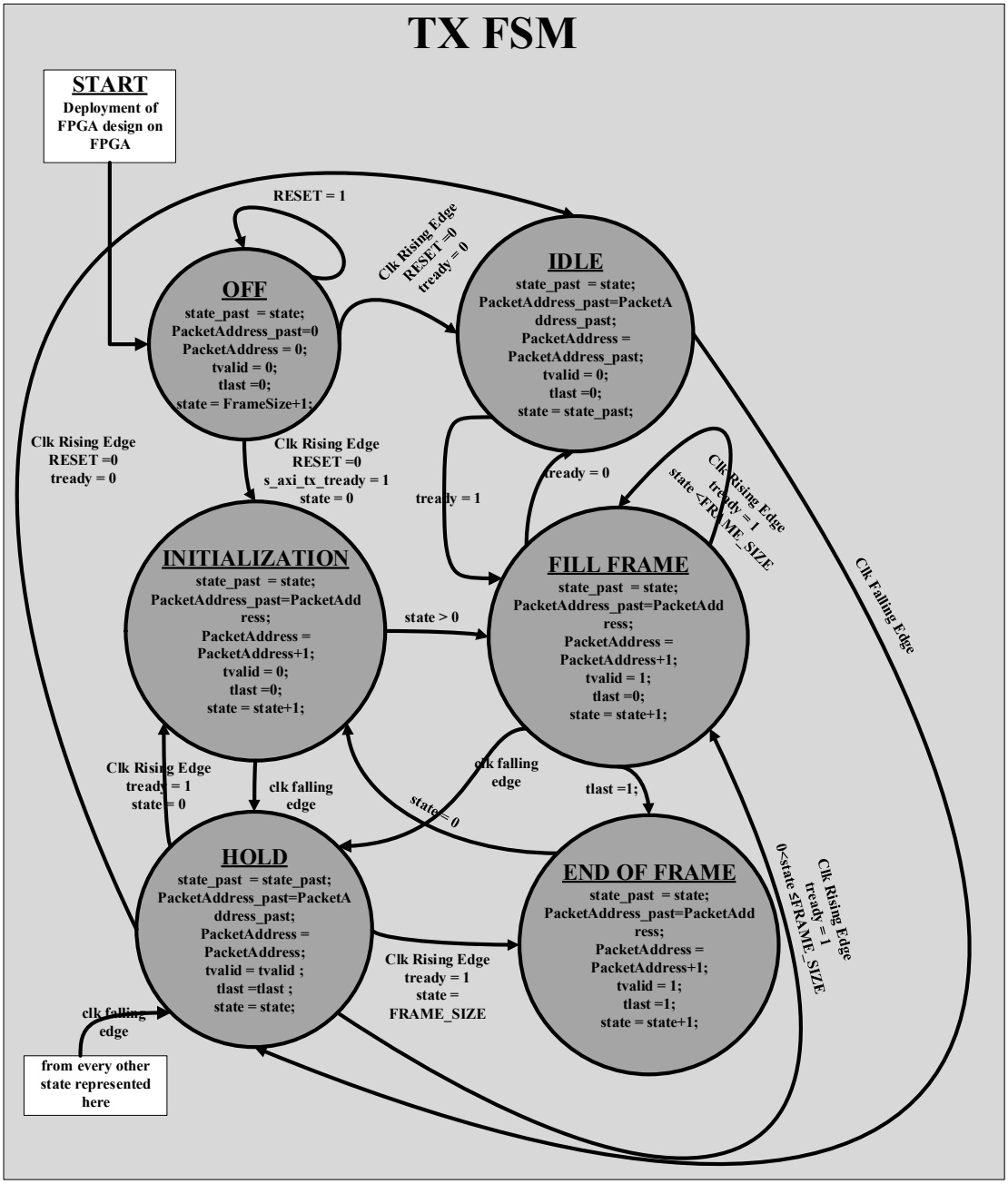


Figure 3.13 The TX FSM , part of the Aurora 64B US+ interface, which takes care of transmitting data to external devices through the Aurora 64B IP Core.

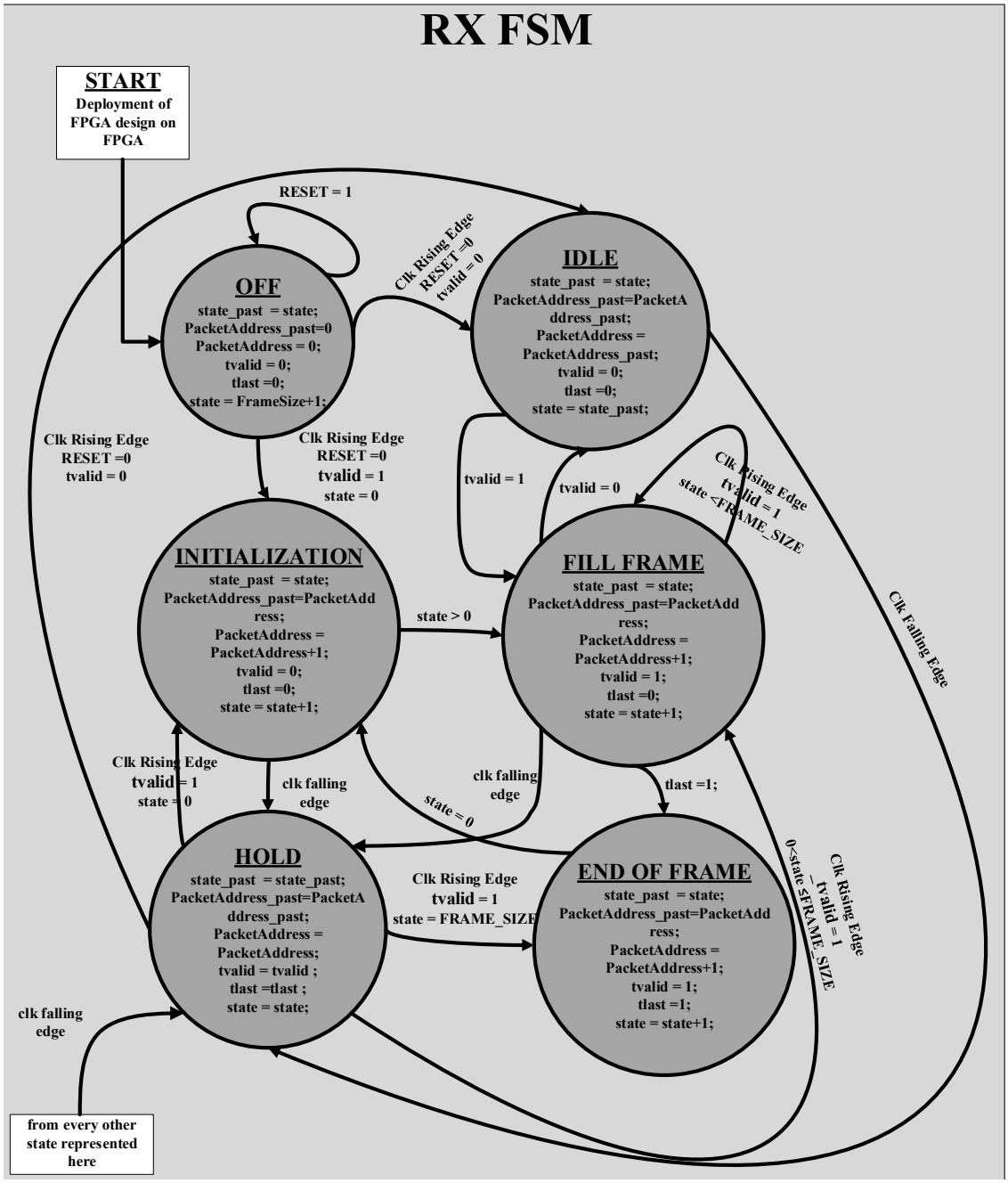


Figure 3.14 The RX FSM , part of the Aurora 64B US+ interface, which takes care of transmitting data to external devices through the Aurora 64B IP Core.

3.2 Parallel Bus Interface

In this section a low latency parallel bus communication interface for high-speed multi-FPGA real time simulation is presented. The interface has been implemented for

communication between two and three US+ FPGA devices. The operation of the interface is -at first- evaluated using a LSFR to compare numerical values exchanged over the bus. Then both the interfaces are used for the simulation of a power electronics system – composed of two dual active bridge converters– using a time step of 70ns. The results of the decoupled simulations are verified against the ones of a monolithic solution running on a single US+ FPGA.

3.2.7 Introduction

The application of HIL techniques on PEPDS -due to the fast dynamics involved- is progressively requiring the employment of FPGA devices for real time simulation execution.

The FPGA devices with their high level of computational parallelism, low computational and I/O access latency allow simulating large PEPDS models with the level of time resolution required by modern power electronics converters.

By applying methodologies such as the LB-LMC [23] it is possible to achieve execution with time step size below 100ns [45][46]. As demonstrated by the scalability tests reported in [45], for a given simulation time step, resource usage scales linearly with the size of the simulation model until reaching the device computational resource usage limits. If the system of interest is too large and cannot be fitted into a single FPGA, multi-FPGA platform must be used. In [24] we defined a methodology for multi-FPGA real-time simulation execution, and we demonstrate its use with a ship power systems example.

To be able to effectively use multi-FPGA approaches for large PEPDS systems the FPGA-to-FPGA communication interface should be flexible, with a reliable synchronization and IO control and fast enough to achieve the desired simulation time step.

The flexibility is related to either the maximum word size, amount of data transferred and connection capabilities to multiple devices. The synchronization of the FPGAs used should also be ensured by the FPGA-to-FPGA communication interface.

A prototype interface of this type has been initially presented in [24] to demonstrate the ND method but presents some limitations in terms of flexibility, speed and synchronization. Specifically, the absence of an external system manager handling the communication synchronization makes impractical to extend the interface beyond two devices and affects the reliability of the synchronization which must be handled by one of the communicating devices acting as master.

Multi-gigabit serial transceivers are commonly adopted solutions for multi-FPGA simulations of large systems such as [47] even though the high amount of clock cycles required for data transmission by the serial interface makes it impractical for high performance RT simulations. The communication latency of those interfaces spans over several hundred of nanoseconds, mainly due to time required for serialization and de-serialization. This is unacceptable for real time simulation when the goal is to maintain the time step size below 100ns.

The parallel bus interface overcomes those issues since data are transmitted over the bus in only one simulation time step clock cycle. This type of interface has found plenty of applications in computer systems over the past years, but the available methods are not suitable for the multi-FPGA RT simulation application proposed here. The well-known Centronics parallel protocol is inadequate due to its directionality and limited data bus size. The Small Computer System Interface (SCSI) is bidirectional, but its bandwidth is limited to few megabytes per seconds and its physical interfaces is not well suitable to modern

FPGA IO interfacing connectors. Parallel bus interfaces as in [48] -being asynchronous, with half duplex data transmission- are not suitable for achieving very low communication latency.

State of the art synchronous solutions such as the IBM PLB CoreConnect [49], would introduce a communication interface too complex for the application discussed in this paper and would not be optimal to implement custom size fixed point data words. The use of solution like Peripheral Component Interconnect (PCI) Local Bus -as described in [50], with a maximum clock frequency of 33MHz- are limited in speed and the performances decrease significantly at the increasing of the number of masters and slaves connected to the bus.

In this paper we introduce a new parallel bus architecture, for point-to-point communication. The proposed solution is flexible, and it employs a data clock of 285MHz. With this interface it possible to achieve simulation time steps of 70ns without introducing any delays between the different model sub-systems. The operation of the interface can be adjusted to different FPGA clock frequencies and IO capabilities and the presented layout -with a single master system manager- facilitates the connection of additional FPGAs, if required by the system of interest

3.2.8 Simulation method description

The proposed parallel bus interface is intended for high-speed RT simulations methods applications so to allow simulation execution across multiple FPGA devices. As mentioned in the introduction the simulation methods of reference are the LB-LMC and the ND method. The ND exploits the LB-LMC features to decouple a system into subnetworks where each of these is attached to NPMs equivalent of the other ones. Let us

assume the original monolithic system has been decoupled in into two subnetworks sharing only one port Fig. 1. A CI as a linear combination of the subnetwork's component sources contributions is used to represent the rest of the system and it is updated at each simulation step, the conductance term of the NPM is vice versa calculated only once off-line. The solution of the original system based on the nodal equation Gx=b is the same as the solution of the decomposed system where each subnetwork is solved based on its own nodal equation indicated in the figure with corresponding subscripts A or B.

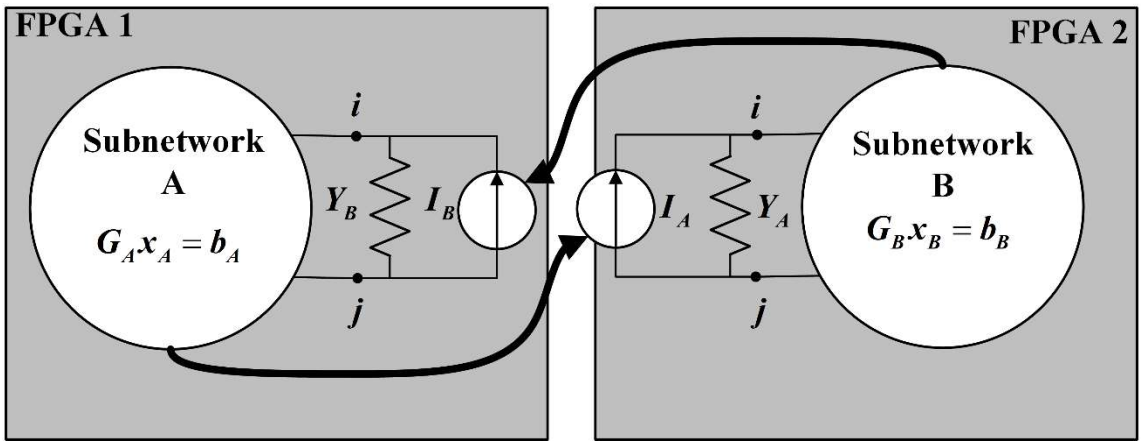


Figure 3.15 Microgrid model nodal decoupled over the multi-FPGA platform.

The simulation method here described has been further extended to allow communication between three FPGAs which for this specific case are US+.

Consequently, The ND method has been further extended to allow the decomposition of the microgrid monolithic system into three subnetworks as illustrated in the diagram of Figure 3.16.

As the diagram shows, differently from the case of two subnetwork, in the present case, the subnetwork B shares a port with each of the other two subnetworks.

In this case the CIs received by A and C are linear combinations Σ of the CI generated by B and the CI coming from the other subsystem.

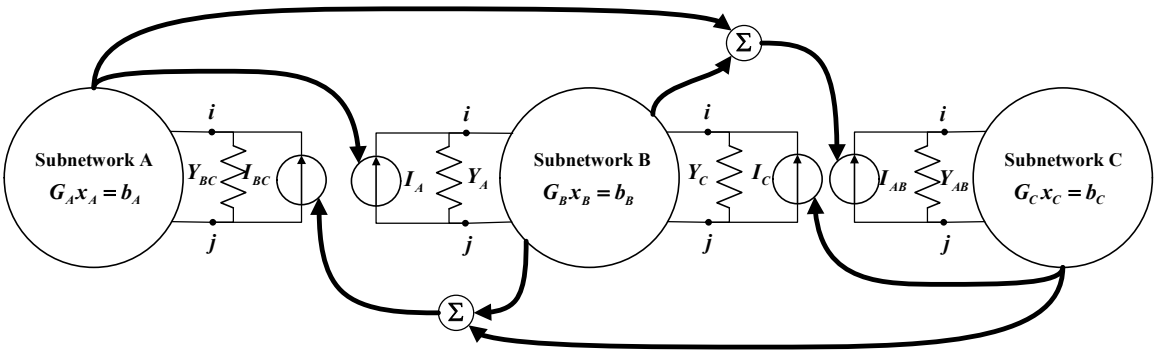


Figure 3.16 Microgrid model nodal decoupled into three subnetworks.

3.2.9 Interface requirements

Let us now proceed analyzing the communication interface functions and requirements.

In general, the parallel bus interface here described can be employed for any point-to-point full duplex communication between FPGA devices implementing FMC connection. In this case each communicating system should only send and receive a set constituted by three signals: one data word of 90bit, one data valid signal for handshaking purposes and one data clock shifted of 90 degrees with respect to the internal one.

The number of FPGA which can be connected with the parallel bus interface is directly proportional to the number of clock and state signals pairs which can be brake out from the system manager board. To ensure synchronization, all the communicating devices should receive in parallel from the system manager those signals.

To minimize clock jitter effects, the state signal synchronizing the communication should be toggling with a duty of 50% with a period twice larger than the period of its clock.

This interface has been mainly designed to realize simulation of system applying the LB-LMC nodal decomposition method. During simulation execution each subnetwork

and so each FPGA- receives through the parallel bus interface, the CI I_A or I_B. To ensure accurate simulation results and maximum execution speed a tradeoff in data representation is achieved with a word size of 90bit. To ensure compatibility with the rest of the design the FPGAs system clock resources and networking must guarantee timing closure over a minimum FPGA design clock value of 3.5ns.

3.2.10 Architecture description

From scientifical point of view, the introduction of the parallel bus increases the scalability of a RT simulator based on a single FPGA by extending the simulation to multiple FPGAs.

To demonstrate this concept and the capabilities of the interface, the parallel bus must be integrated in a multi-FPGA platform design. A block diagram illustrating such system hosting the parallel bus interface for communication between two US+ devices developed for this work is illustrated in Figure 3.17.

In order to ensure communication synchronization a system manager design delivers in parallel to the communicating devices FPGA1 and FPGA2 the main clock "main_clk" and the state signal "state" which following a leap frog scheme illustrated in Figure 3.18 toggles between the simulation engine and the parallel bus communication.

Each one has an execution time step of 35ns, based on sim_clk, realizing an overall simulation time step of 70ns.

The main_clk with period equal to 3.5ns is used by each FPGA device clock manager to derive the simulation engine execution clock "sim_clk" (the same generated in Virtex) used to synchronize the simulated subsystem and its control. The sim_clk has a period of 35ns to realize the execution time interval of the 70ns simulation time step.

The main clock is also used to derive the data clock (data_clk) and its 90° version (data_clk_fwd) which are low jitter clocks employed to exchange data over the parallel bus both with period of 3.5ns.

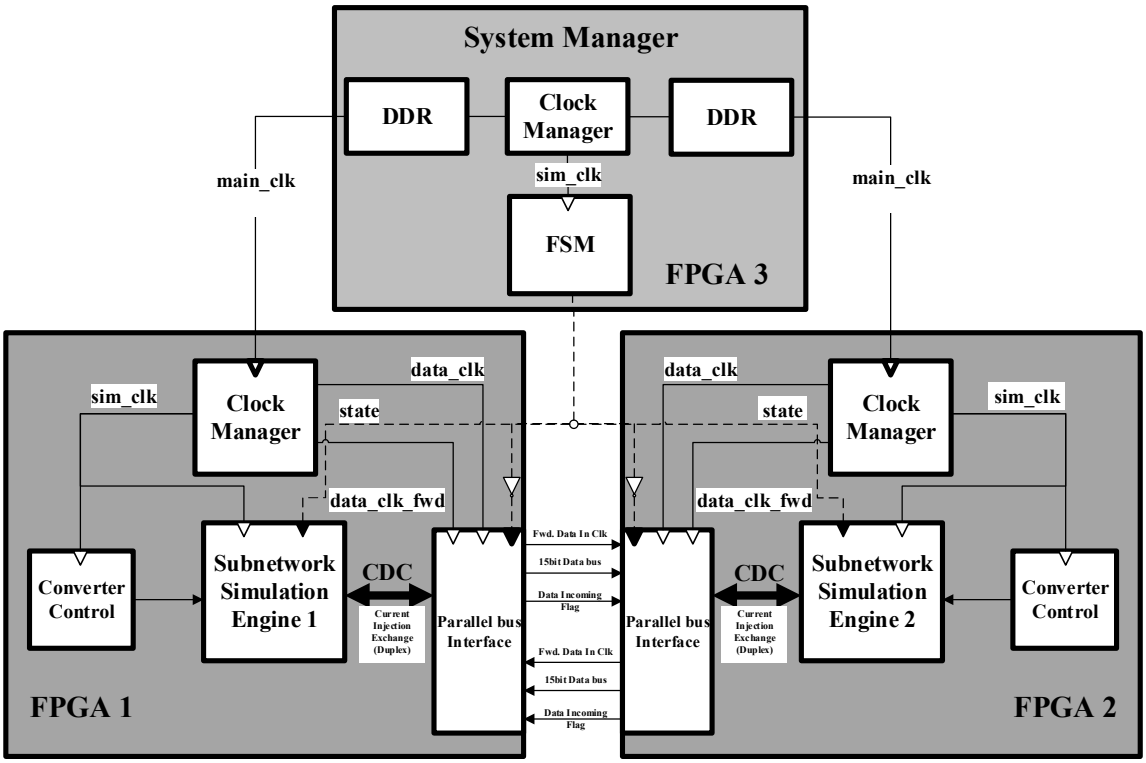


Figure 3.17 Simulator layout of a parallel bus interface connecting FPGA1 and FPGA2 with FPGA3 acting as system manager.

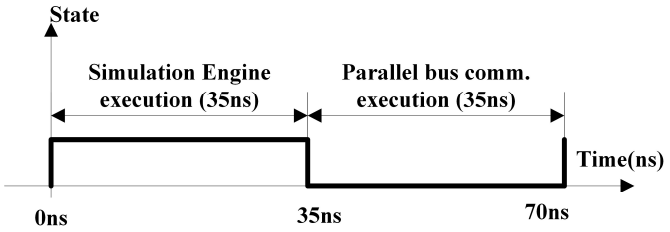


Figure 3.18 Time diagram of the leapfrog FSM state signal.

A CDC interface with corresponding timing constraints is needed to correlate the simulation and parallel bus interface clocks to avoid metastability issues. For this purpose two-stage synchronizer registers have been employed (also FIFOs are suitable).

The architecture described so far, employing the same type of interface has been extended for communication between three US+ FPGA devices as shown in Figure 3.19.

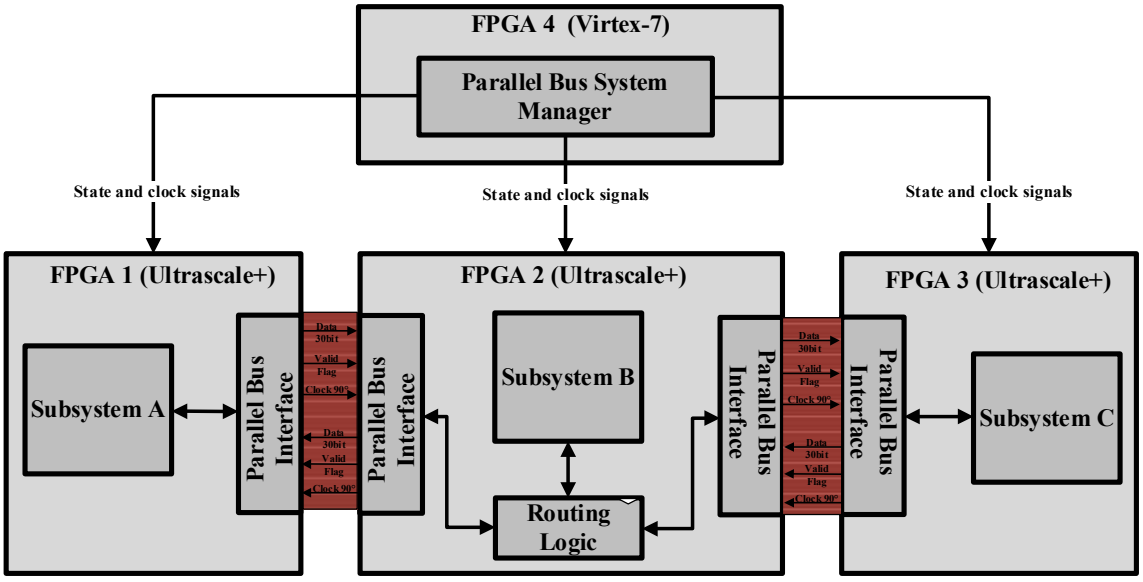


Figure 3.19 The three-FPGAs based parallel bus architecture.

The most noticeable difference in terms of FPGA design, is the presence of the routing logic needed to realize the CI exchange between the subsystems (or subnetworks) A and C which generated CIs are transmitted to B and then linearly combined with the CI generated by B before to be sent out. This operation. is illustrated in more detail in Figure 3.20.

The routing logic operates according to an internal FSM which upon reception of valid and ready signals linearly combines the internal CI of the subsystem B namely I_B respectively with $I_{A,}$ to generate the CI named I_{AB} which is transmitted to the subsystem C, and $I_{C,}$ to generate the CI named I_{BC} which is transmitted to the subsystem A.

The system manager enhances the platform scalability flexibility -in terms of introduction of new communicating devices receiving in parallel from the system manager states and clock signals. The communication is operated in full duplex. The execution and

communication time steps both are equals to 35ns for an overall simulation time step of 70ns. The maximum number of communicating devices depends on the IOs capability of the system manager while each additional subnetwork introduced withing the diagram of Figure 3.16 will require a routing logic like the one of B to exchange CIs with the adjacent subsystems.

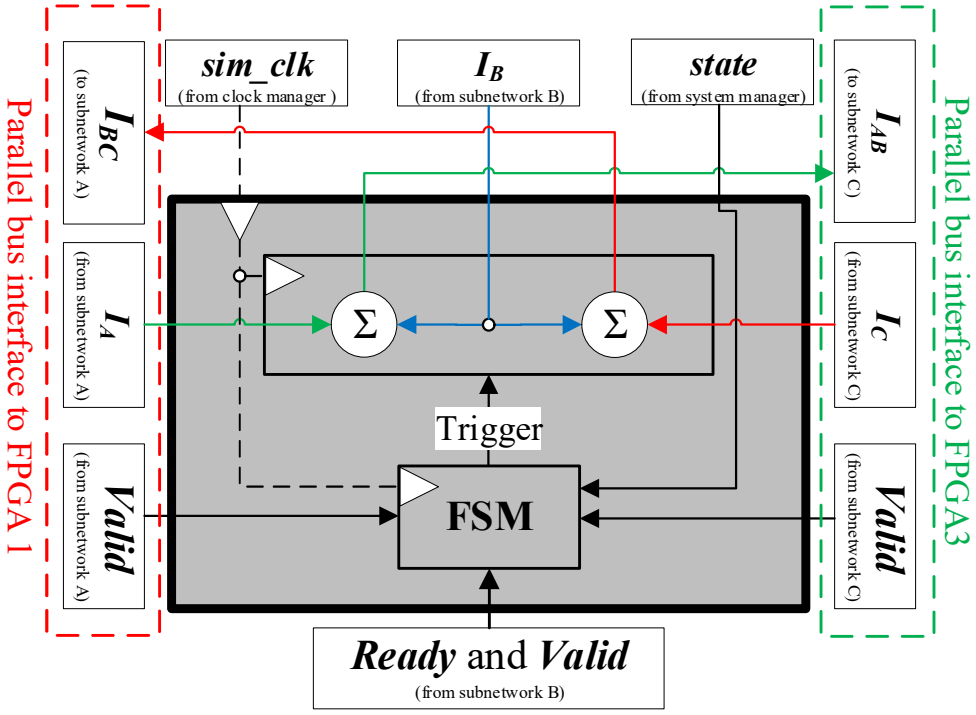


Figure 3.20 Block diagram of the routing logic within the parallel bus interface for three devices.

3.2.11 Parallel bus internal structure

To clarify the operation of the parallel bus interface, here it is provided a technical description of the internal architecture of the parallel bus module integrated within each FPGA design.

In Figure 3.21 the block diagram showing the interfacing IO connections with the external world and internal structure of the parallel bus interface FPGA design is reported with detail about packing and unpacking of the 90bit data word.

The data_clk synchronizes the FSM of the sending side of the interface (TX FSM). The data signal exchanged over the bus is a custom fixed-point word which can host up to a 90bit word size. This is serialized in three 30bit sub words sent out in parallel through ODDR DATA that is an Output Dual Data Rate (ODDR) register, synchronized by data_clk to the FMC ports which are connected to the other communicating device through an FMC ribbon cable which is the physical support of the parallel bus. A valid flag signal "Valid Flag" and a phase shifted version of the clock "data_clk_fwd", with nominal phase shift equal to 90°, are forwarded to the other device with dedicated ODDRs.

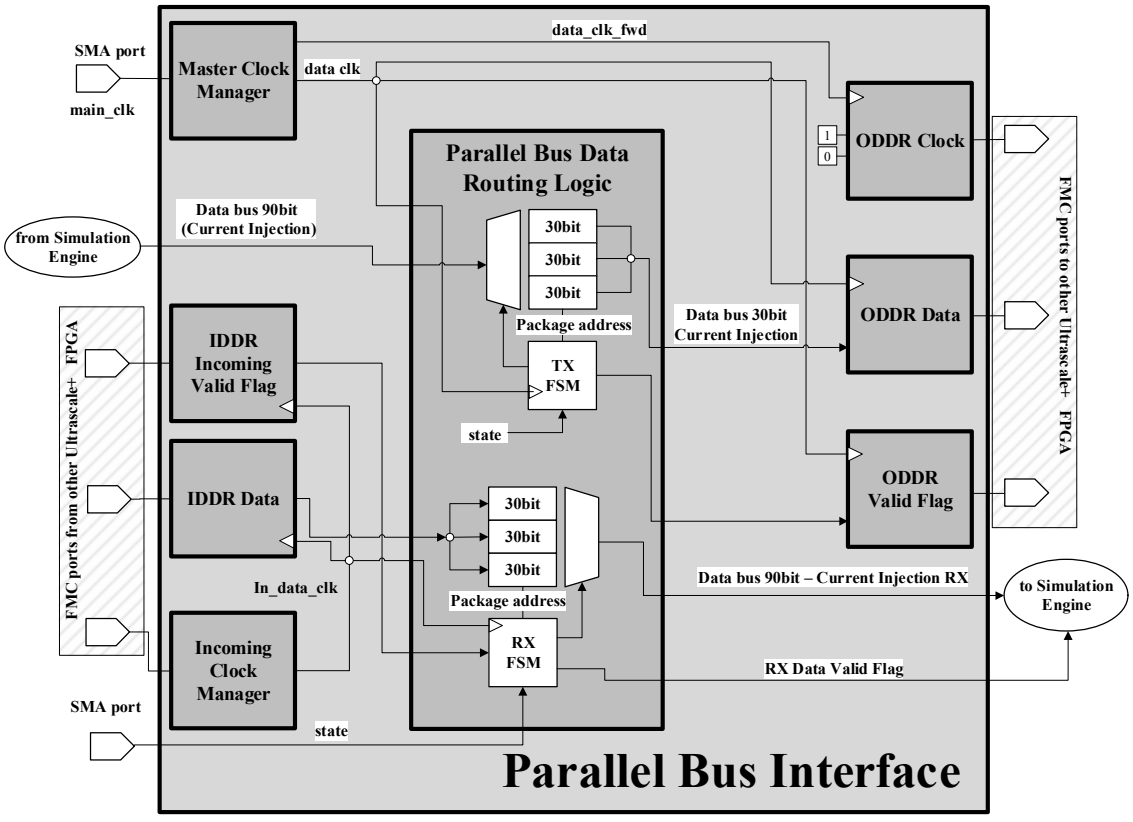


Figure 3.21 Internal design of the parallel bus interface.

All the output signals mentioned here are the same type of the signal received (RX) in the FPGA from the other device. Specifically, a clock manager namely "Incoming Clock Manager" is sourced by "in_data_clk_fwd"-the incoming data_clk_fwd from the other

device- and it is used to synchronize the operations of IDDR Data, that is an Input Dual Data Rate register and the receiving side FSM (RX FSM). The received data are handled using data_clk_fwd to stabilize them on the receiving ports. Basically, the phase shift ensures a waiting time during which the electric impulse can stabilize before to be used by the internal logic of the FPGA. The IDDR Data is used to receive the 30bit word coming from the other FPGA. The RX FSM selects only those sub words which are valid, according to the Incoming Data Flag, then will pack them in a 90bit word which is sent to the simulation engine running on the same FPGA outside the parallel bus interface module.

3.2.12 Laboratory setup

In this paragraph the laboratory setup employed to test the newly developed parallel bus interfaces is described. To achieve this goal, a multi-FPGA simulation platform which setup is illustrated in Figure 3.22 has been developed.

In accordance to Figure 3.17, the Virtex-7 VC707 FPGA evaluation kit is employed as system manager while two US+ FPGA evaluation boards, namely Ultrascale#1 and Ultrascale #2, kits communicate through the parallel bus blue cable (P. bus cable) connected on the FMC connector J2 FMC HPC1 type.

The platform offers the possibility to log data in RT on oscilloscope through a TI-DAC34H84 connected via FMC flexible cable to connector J22 of the Ultrascale#1.

Being LPC, the flexible cable covers only the LPC subset of the whole FMCP HSPC J22 connector. Additionally, the fiber optic based QSFP and SFP cables connections including the FM-S14 card are meant for interactions with external devices. In this specific case the fiber optic cable connected to the single SFP socket of the Virtex-7 transports the signal which allows to trigger the parallel bus communication from a user console running

on a CPU based machine which is connected to the platform via a USB hub using the JTAG protocol. Specifically, the user can trigger the parallel bus interface from its CPU machine using a user console developed on RT-LAB and running on Opal-RT OP5607 communicating with the multi-FPGA platform via Aurora 8b10b protocol employed to send the trigger signal in simplex to the parallel bus FSM running on the system manager on Virtex-7.

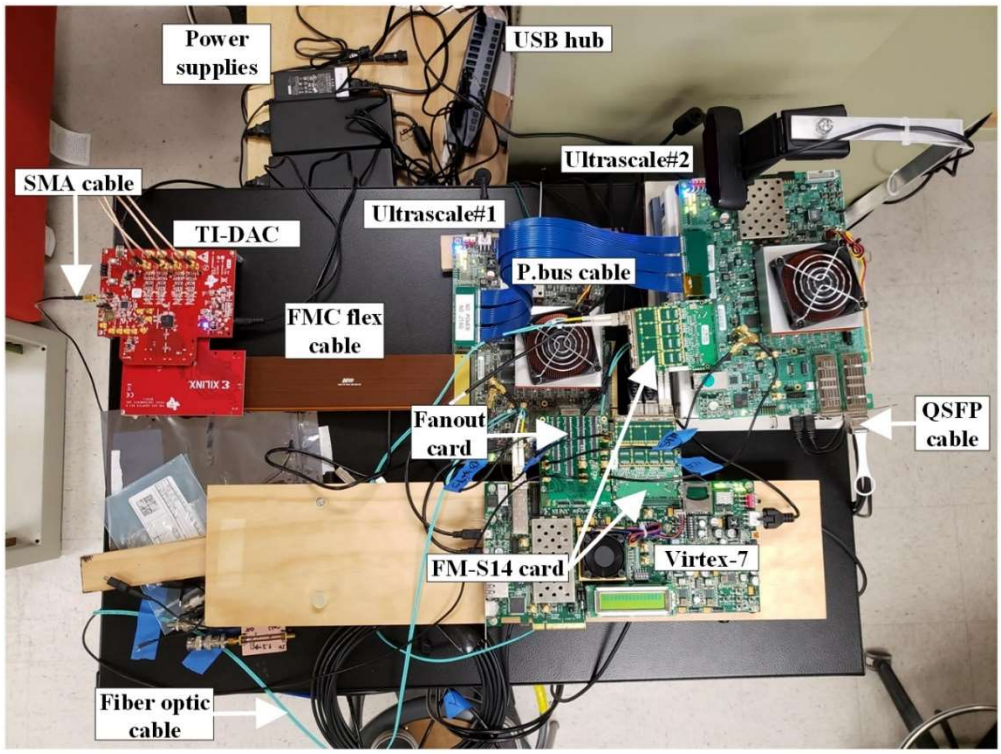


Figure 3.22 The multi-FPGA setup employing the parallel bus interface for two communicating FPGA devices.

To increase scalability and flexibility of the platform and make it suitable for future expansion, single ended data and clock signals are break out from the Virtex-7 board through a custom Fanout FMC card and delivered in parallel from the Virtex-7 to both US+ devices through SMA shielded cables.This setup has been then extended to host a third US+ device and the overall lab setup is illustrated in Figure 3.23.

In both configurations, the setup requires the FMC ribbon flexible cable to connect the TI-DAC to Ultrascale#1 to avoid mechanical interferences between devices hardware boards.

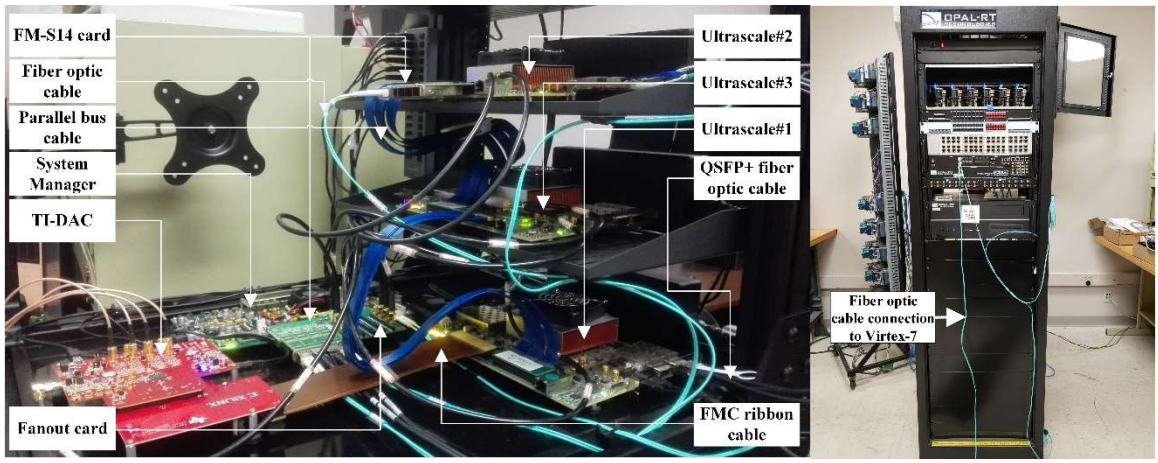


Figure 3.23 The lab setup employed to test the parallel bus interface with three communicating US+ FPGA devices.

3.2.13 Simulation platforms FPGA design description

In this section the simulation platforms employed to test the parallel bus interface are described from the FPGA design point of view. Each design has been realized in VHDL language using the VIVADO tool chain.

A single-FPGA based platform has been developed to simulate the monolithic system used as a test reference and a block diagram showing its internal FPGA design is illustrated in Figure 3.24

By employing the LB-LMC method described in [45] the simulation engine containing the model described into the next sections is simulated with its PSC generating gates signals with a sim_clk equal to 70ns. The generated data is sent through CDC to the DAC interface which implementation is described in [46]. This interface allows to log data on the oscilloscope employing a DAC device TI-DAC34H84. The DAC is sourced by a

70ns reference clock (dac_ref_clk) generated by the system manager design running on Virtex-7 and it generates a replica of the same clock (dac_clk) which synchronizes the DAC interfaces when gathering data from the simulation. To stabilize the signals coming into the TI-DAC a 90° version of the clock (dac_clk_90) is employed.

Is worth to mention that beyond parallel bus testing, this platform can be employed for RT-HIL simulation running simulation engines of PEPDS down to a simulation time step of 35ns while in the present case a simulation time step of 70ns is employed only for comparison of the results from the multi-FPGA platform employing the parallel bus illustrated as well in this section.

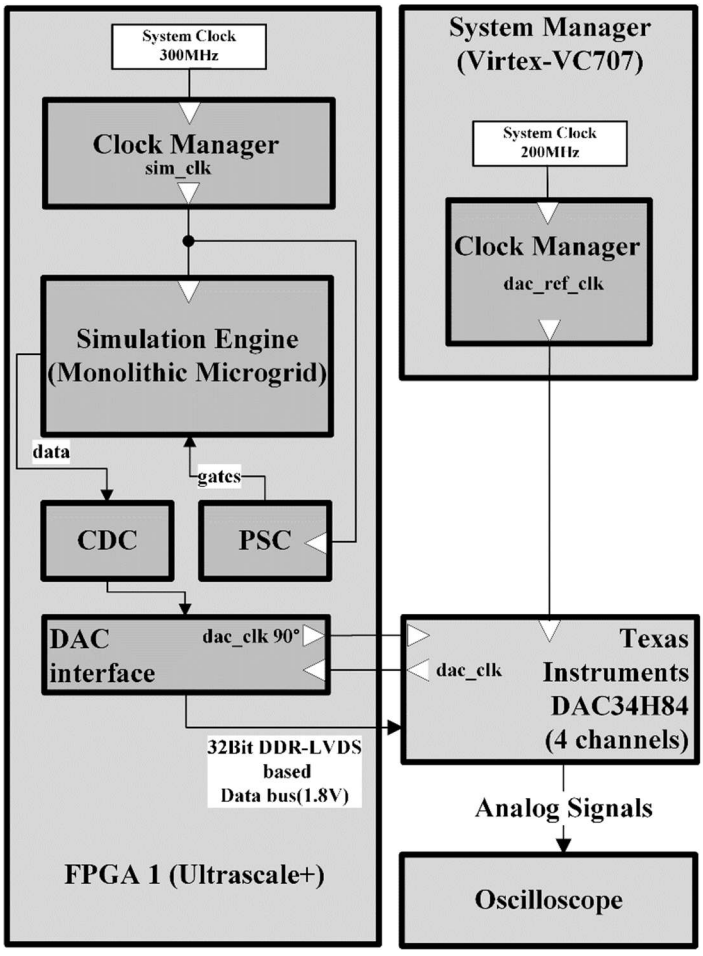


Figure 3.24 Single FPGA RT-simulation platform.

A detailed schematic of the aforementioned multi-FPGA simulation platform hosting the parallel bus interface for two communicating devices and based on the general architecture described in Figure 3.17 is illustrated in Figure 3.25.

The parallel bus communication is triggered by user console on host-PC connected via Aurora8b protocol and fiber optic cable to the system manger.

For simplicity only one DAC device is embedded in this layout which interface is integrated within the FPGA1 design. In this schematic it is highlighted the distribution of the state signal to either the simulation engine and the parallel bus interface on which it is negated to realize the leap frog operation.

Regarding the clocking, differently from the platform described in Figure 3.24, in order to ensure data integrity and time closure within the overall platform design, the sim_clk is derived within each US+ device from the main_clk externally generated by the system manager on Virtex and not anymore internally from the US+ system clock at 300MHz.

Finally the block diagram of the simulation platform FPGA design hosting three US+ FPGA devices is illustrated in Figure 3.26. The design of FPGA2 and FPGA3 is similar to the ones of FPGA1 and FPGA2 in Figure 3.25 while the central FPGA design, , namely FPGA1, is more complex since contains the routing logic to support the parallel bus communication for three FPGA meant for LBLMC nodal decomposition applications of PEPDS. It is highlighted the CDC needed between the simulation engine and parallel bus interface clock domains which specially in this complex design covers a fundamental role for timing closure. The system manager design running on Virtex-7 must provide in three pairs of state and clock signals to each of the three communicating FPGAs.

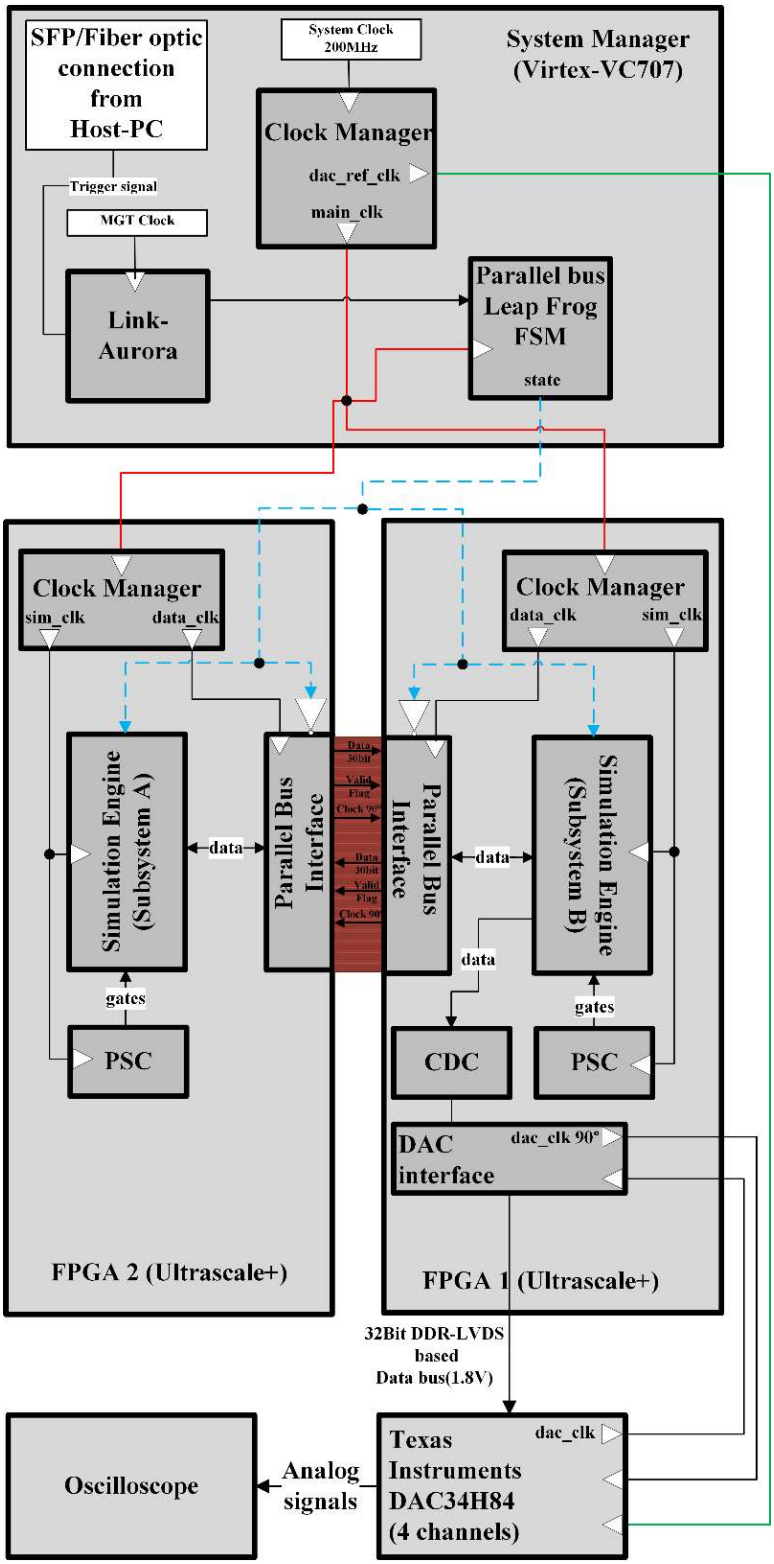


Figure 3.25 A detailed schematic of the simulation platform illustrating the FPGA designs and the main connections.

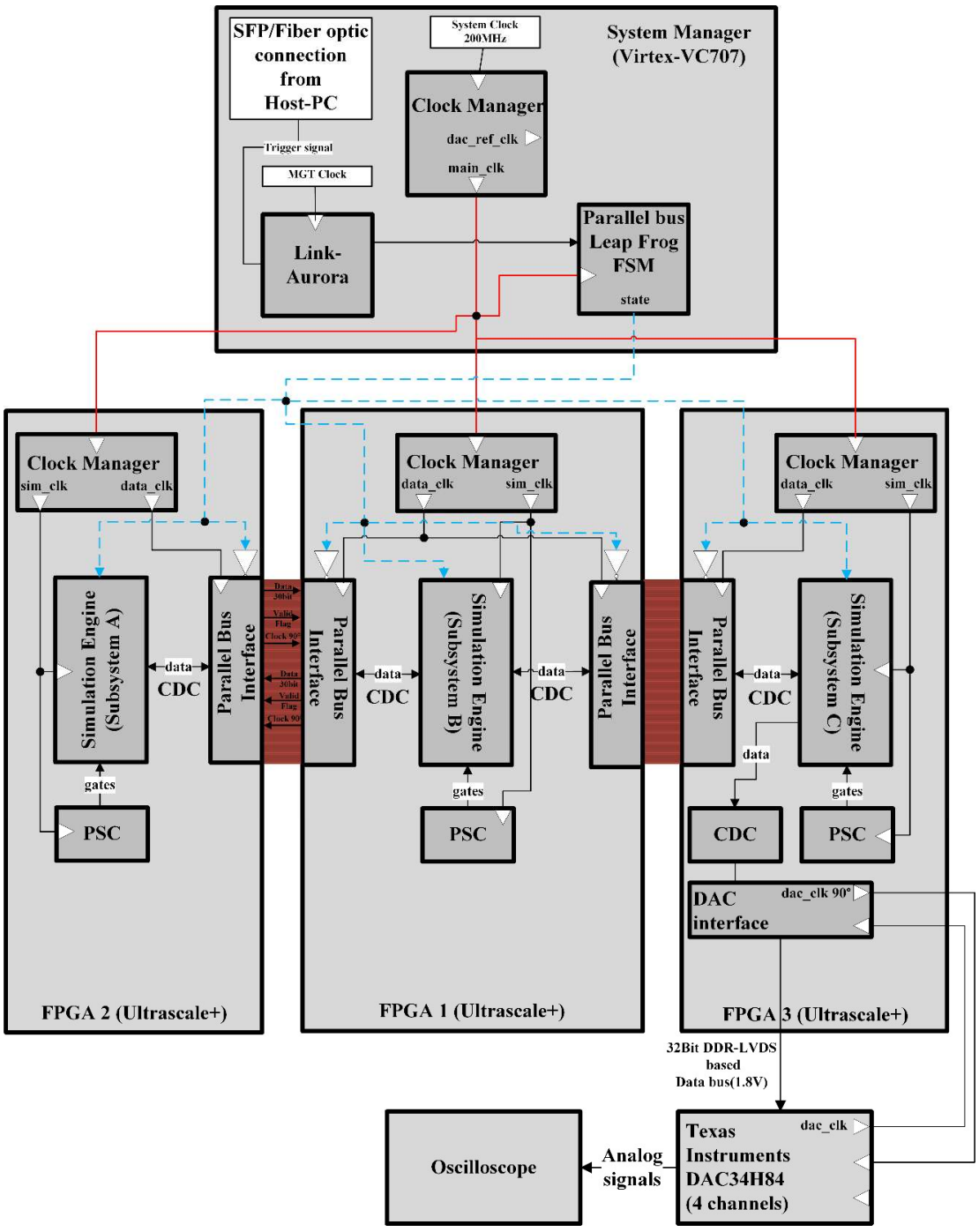


Figure 3.26 Block diagram of the multi-FPGA platform FPGA design for three US+ devices.

3.2.14 Results

The developed parallel bus interface communication has been tested by having both US+ devices exchanging in full duplex 90bit fixed point data words generated by the

simulation engine depicted in Figure 3.17 included in each FPGA design and having them comparing the transmitted and received data for error detection and counting. The parameters adopted for this communication test are illustrated in Tab. I. For timing closure of the FPGA design, the sim_clk period is an integer multiple of the data_clk.

Table 3.1 Parallel bus interface main communication clocks parameters

FPGA design signal	Desrcription	Value
data_clk	Data communication clock frequency (MHz) and period (ns)	285MHz, 3.5ns
sim_clk	Simulation engine execution clock period	35ns

Each US+ simulation engine, following a deterministic process, generates numerical values using a Linear Feedback Shift Register (LSFR) during the simulation cycle (duration of 35ns) and exchanges them with the other device during the communication cycle (duration 35ns).

Then at the beginning of the next simulation cycle, each FPGA design compares its previously internally generated number with the number received from the other device and if the equality is satisfied, the “error detector” signal remains low (zero) and the error counter does not increase.

For the purposes of the test realized for this work, data have been logged through JTAG and USB connection on the user workstation connected to the platform through the USB hub depicted in Figure 3.22.

The communication test results, monitored with the Vivado Integrated Logic Analyzer (ILA core) are, shown in Figure 3.27 where the signals appearing in the left column are described here below:

user_sma_state_debug:

It is the parallel bus FSM's state signal which when high commands the simulation execution (numbers generation) and when low commands the parallel bus communication execution. Both simulation and communication cycles are each one executed in 35ns.

data_out_debug:

this is the output of the simulation engine consisting into the random number internally generated which on the next simulation cycle will be compared with the incoming signal from the other FPGA named data_in_debug;

data_in_debug:

incoming random number from the other FPGA;

ErrorDetector_debug:

this flag is raised by the simulation engine if data_out_debug is different from data_in_debug.

ErrorCounter:

this is a counter which accumulate the ErrorDetector_debug's values to keep track of previous errors.

Figure 3.27 Communication test results from data logger during FPGA execution.

With reference to Figure 3.27, the signal user_sma_debug corresponds to the state signal depicted in Figure 3.18. On the falling edge of the state signal, the simulation engine has computed in one time step equal to 35ns a new value corresponding to data_out_debug

which is sent out to the other device within a time step of 35ns until the next rising edge of the state signal.

At the same time, since the communication over the bus happens in full duplex, a new incoming value namely data_in_debug is transmitted over the bus from the other device to the current device and it is read by the simulation engine on the rising edge of the state signal.

At this instant, which for example can correspond to the sample 2003 indicated by the yellow flag in Figure 3.27, if data_in_debug is equal to data_out_debug then the ErrorDetector signal remains low and the counter ErrorCounter_debug does not increase.

The results show how thanks to the parallel bus interface, a simulation can be executed within an overall simulation timestep of 70ns were only 35ns are required for communication without inserting any further time delays.

A more application-oriented test has been realized by simulating in RT on the single FPGA based platform described before a monolithic microgrid model for electric ship applications applying the LB-LMC method and comparing its simulation results with the ones obtained by simulating the same model nodal decomposed on the multi-FPGA platform based on the parallel bus interface.

The network model considered for this work, based on Dual Active Bridge (DAB) converter model has been implemented exploiting the ORTiS tool deeply described in [28] [51] [52]. The library provides a full switching state space model of the DAB converter which circuit is shown in Figure 3.28 and which development has already been described in Chapter-2. Electrical and modulation parameter of such model are listed in Table 3.2. A microgrid model based on two DAB converters has been realized and decomposed in two

and three subnetworks as illustrated in Figure 3.29 and Figure 3.30. Those are the test cases for the parallel bus interface for two and three communicating devices.

Table 3.2 Simulation models parameters.

Parameter's ID	Description	Value
V_g	DC Generator nominal voltage	12kV
V_0	DC Load nominal voltage	1kV
R_g	DC Generator resistance	1mΩ
R	Cable resistance	1mΩ
L	Cable inductance	1μH
R_{11}, R_{22}	Capacitors ESRs	0.8Ω
C_1, C_2	Capacitances	540μF
L_1, L_2	DAB Transformer inductances	150μH
N	DAB Transformer coils turns	12
f_{sw}	Switching frequency	60kHz
φ	PSC phase shift	60°

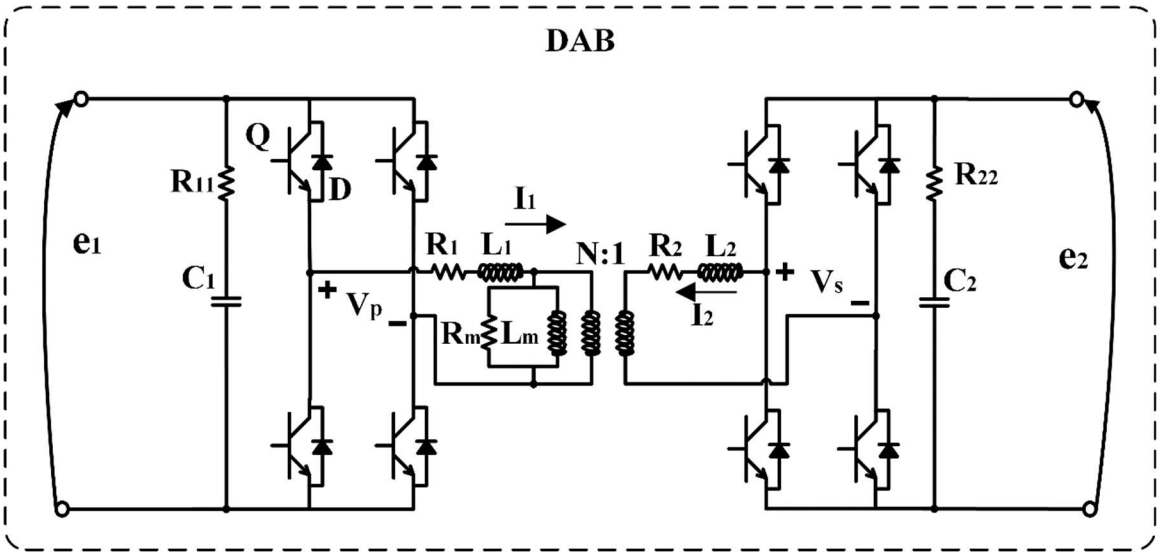


Figure 3.28 Circuit of the DAB state space full switching model.

With reference to Figure 3.29 and Figure 3.30, the DAB #2 waveform have been logged for either the monolithic (continuous line) and decomposed (dashed and dotted lines) solutions to realize the comparison to easily evaluate and visualize the effectiveness of the parallel bus interface.

Nodal decomposed solutions are indicated in the plots respectively with subscripts “dec2subs” and “dec3subs” while the remaining without subscripts refers to the monolithic one.

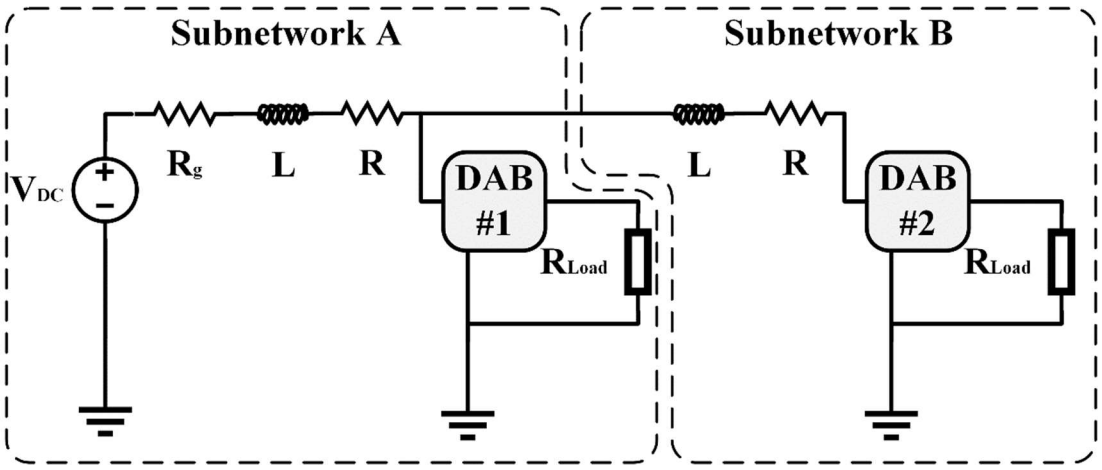


Figure 3.29 Microgrid model nodal decomposed into two subnetworks.

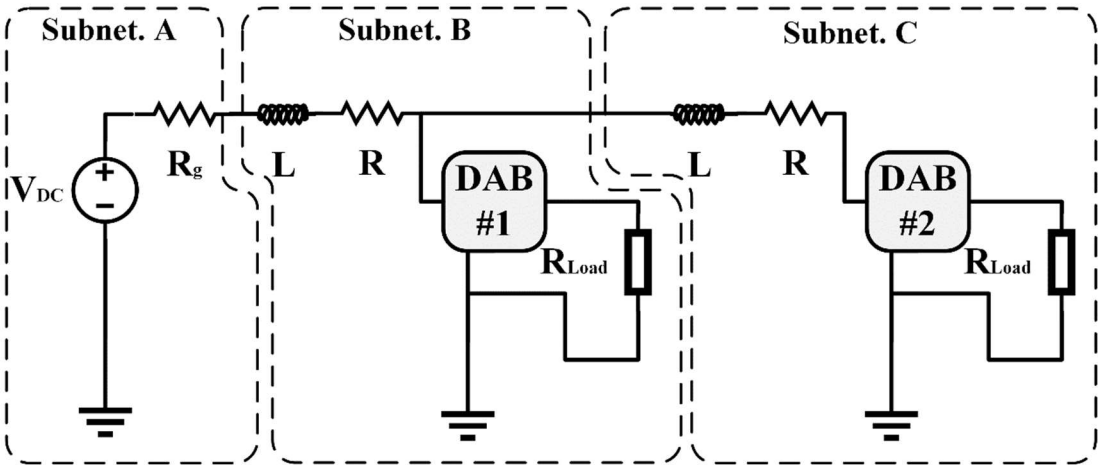


Figure 3.30 Microgrid model nodal decomposed into three subnetworks.

In Figure 3.31 are presented the results from the RT simulation on the multi-FPGA platforms described previously of the nodal decomposed microgrid model respectively in two and three subsystems.

The quantities considered, are the voltages of the primary and secondary side of the DAB’s transformer, namely V_p and V_s and correspondent currents I_p and I_s.

Besides negligible differences introduced during post processing to filter random noise generated by the measurement system, when logging data on the oscilloscope, the waveforms of both solutions match perfectly to each other, with negligible error, proving the correct operation of the multi-FPGA platform and the parallel bus interface.

This test demonstrates how the newly developed parallel bus interface is suitable for highly accurate system level RT-HIL simulations involving systems of considerable size.

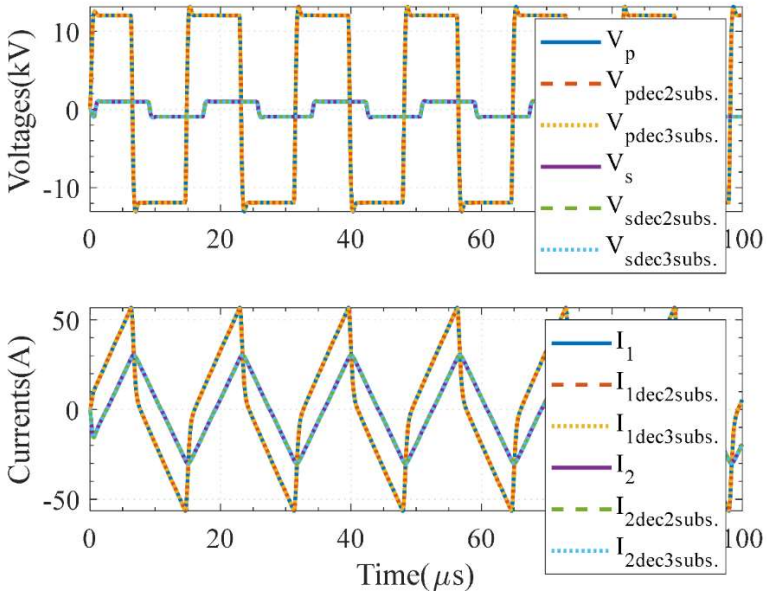


Figure 3.31 Results of the RT simulation on the multi-FPGA platforms of the microgrid model nodal decomposed in two and three subnetworks respectively with subscripts “dec2subs” and “dec3subs”.

3.3 Conclusions

In this chapter the implementation of parallel and serial communication interfaces necessary for the realization of the RT-HIL methodologies proposed in this work has been presented. The serial interfaces based on the Aurora protocol are meant for application where a certain latency is tolerated such as the application of the CHIL technique and the

co-simulation involving multi-rate systems. For this purpose an Aurora interface with 64bit encoding has been developed. In the case of the co-simulation application, the interface has been extended with a link design running on Virtex-7 to facilitate the communication with the OPAL-RT platform.

To facilitate and make more agile HIL testing of systems with multiple MMC converters, it has been developed a protocol-based interface used to substitute traditional analog measurements and digital gate signals. The interface clearly introduces additional delays; we have analyzed this in detail. Through various tests we have shown how those delays are responsible for an error at max equal to 2.8%.

To answer the increasing demand of more accurate and scalable RT simulations of complex PEDPS employing fast switching devices, a novel parallel bus architecture is presented in this work. This new prototype is flexible, fast, scalable and does not introduce any communication latency within the given simulation time step of 70ns as demonstrated by the test results presented. The interface flexibility and scalability is increased by the introduction of the external system manager which easily allow to extend the simulation platform to multiple FPGA devices keeping the same level of performance. The high level of accuracy is guaranteed by the possibility of employing a custom data word for fixed point computations up to 90bit and by the communication frequency of the bus, beyond 285MHz ($\leq$3.5ns) which guarantees no communication latency during simulation.

The developed interface has been successfully tested either monitoring the data traffic generated by a LSFR deterministic register and by comparing results from a RT-simulation based on the LB-LMC method of monolithic ad nodal decomposed PEPDS based on DABs converters for ship applications.

CHAPTER 4

HARDWARE-IN-THE-LOOP TESTING OF HIGH SWITCHING FREQUENCY POWER ELECTRONICS CONVERTERS

4.1 Introduction

The results described in this chapter already published in [1], introduce the basis of the methodology developed for this dissertation research to perform RT-HIL experiments on PEPDS . Specifically, it is described how using the LB-LMC method it possible to perform RT-HIL experiments on a FPGA based platform with small time steps. The system simulated is a micro grid model composed of state space full switching converters models implemented with the same methodology illustrated previously. Specifically, it is simulated a notional ship power system where one converter is externally controlled using the developed HIL simulation platform. The ship system is composed of eight converters all operating at 100 kHz, the real-time simulation is performed using a FPGA-based platform and a 50 nanosecond simulation time step. The RT-HIL communication interfaces is analogic and based on hard wiring.

4.2 Literature review

Future electric ship power systems are expected to involve an increasing number of high switching frequency power converters with a switching frequency that can reach ≥100kHz. In recent years it has been observed that testing these devices close to application conditions is a critical step.

Real-time simulation is a fundamental enabler foradvanced testing techniques like Hardware In the Loop (HIL) and Power Hardware In the Loop (PHIL); for this purpose a few commercial products are available on the market.

Many of these commercial real-time simulators can systematically perform simulations at system level with time steps varying from 1 to 50µs. Nevertheless, while commercial solutions are available, real-time simulation with a time step smaller than 1 µs is still a research field in need of fully defined methodologies for parallelization, code generation, and scalability. Furthermore, it still often requires significant tweaking by the user. In this context several papers have been published in recent years looking at FPGA-based pure [53][54]or co-simulation [13][55] platforms for the execution of under 1µs real-time simulation.

In [1] it was presented a FPGA implementation of the Latency-Based Linear Multi-step Compound Method (LB-LMC) introduced in [23] which can perform real-time simulation of high frequency switching power electronic systems with time steps of 50ns and below. With such fidelity in time step resolution, this approach proves to be suitable for HIL simulation where external high frequency controllers are introduced into a closed loop with the simulated system.

In this work, we explore this suitability by developing a HIL setup where a switching converter with a large power system simulated on a FPGA via LB-LMC is controlled externally in a closed-loop configuration by a microcontroller operating at frequencies of 100kHz.

An overview of the simulation method used is presented, followed by how this method is implemented on a FPGA.

Then, the system test model used in the experiment is discussed. Afterward, the HIL setup and implementation are discussed. Finally, results from the experimentation are presented.

4.3 Test Model

As a testing scenario, we use the dual-bus notional shipboard power system displayed in Figure 4.1. The system is composed of six DC/AC converters and two DC/DC converters. Parameters for this system are set so to obtain 40MW of total load and the DC/DC converters are set to output 12kV DC voltage onto bus lines. The overall system has 54 nodes, and 96 switches.

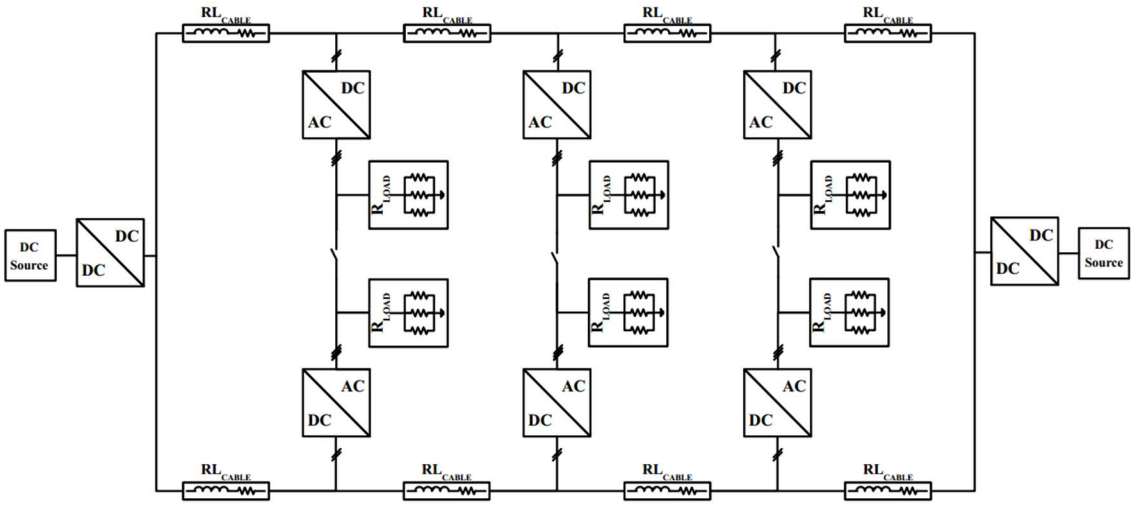


Figure 4.1 Dual-Bus Notional Shipboard Power System.

Five of the DC/AC converters are controlled internally by an open-loop PWM-based controller which has the converters produce 3-phase AC output with frequency of 60Hz.

The open-loop controllers are set to operate with 100kHz switching frequency. Similarly, the DC/DC converters are controlled in open loop with 100kHz switching frequency.

For HIL simulation, one of the DC/AC converters in Figure 3 is configured to operate with a closed-loop controller which is implemented externally from the model and FPGA that simulates said model. The 3 phases AC voltages of this converter are converted to analog signals via a digital/analog converter, so to be sampled by the external controller. Moreover, this converter is configured to take three switch control signals provided by the external controller.

4.4 Hardware-in-the-Loop test bench

In this section, we discuss our test bench platform for HIL simulation of our test model. We begin by discussing the overview configuration of our setup, followed by how the FPGA design is implemented for HIL interfacing, and conclude by discussing the controller implementation.

4.4.1 Overview

For system simulation, a Xilinx Virtex-7 FPGA VC707 evaluation kit has been used. To generate external analog representation of system states to control, a four-channel digital/analog converter evaluation module based on a Texas Instruments DAC34H84 16-bit DAC was used. So those analog signals can be received as input, a four-channel analog/digital converter evaluation module based on a Texas Instruments ADS4449 14-bit ADC was also used. For the control of the system simulated in real-time, we used a Texas Instruments Delfino F28335 microcontroller evaluation board. Due to the differences in voltage level between the simulation FPGA platform and controller (1.8V and 3.3V, respectively), and the voltage level of the analog I/O modules (0.5V), a level shifting adaptation board has been developed and used between these devices. In Figure 4.2, a block diagram representation of the test bench system composed of these parts is displayed.

The physical lab setup of the test bench is shown in Figure 4.3.

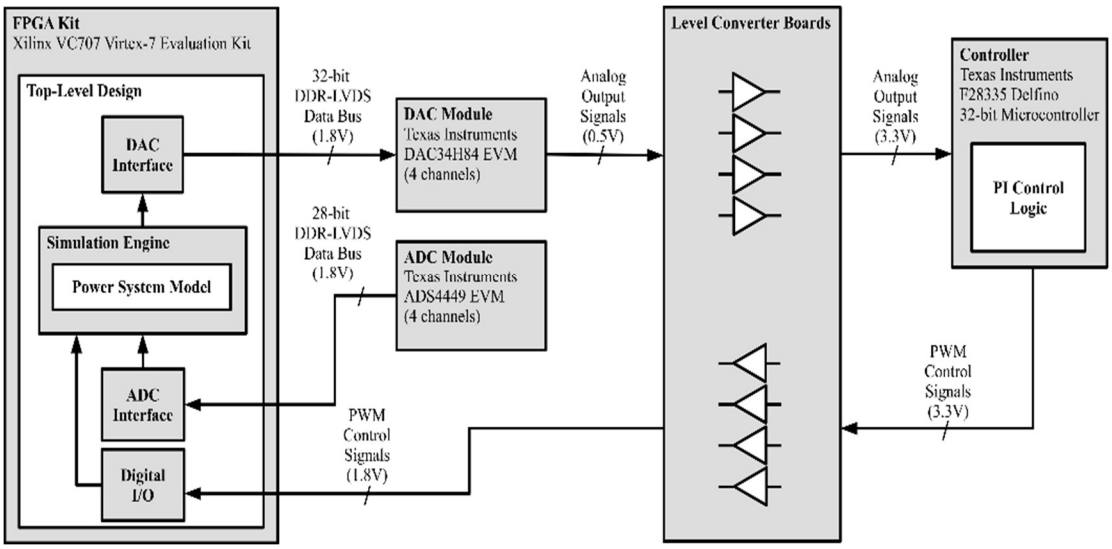


Figure 4.2 HIL Test bench scheme

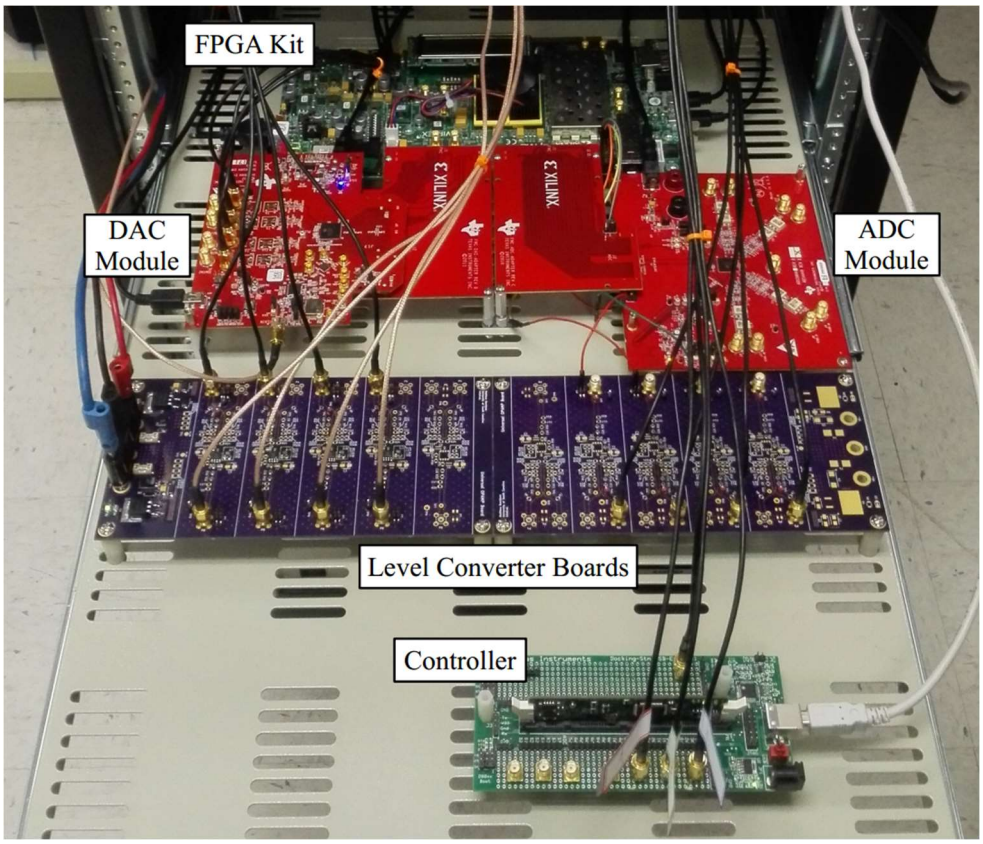


Figure 4.3 HIL Test Bench Lab Setup.

4.4.2 FPGA Design with DAC Interfacing

As shown in Figure 4.4, a top-level design is developed for the FPGA that constitutes the logic for the HIL simulation. The core of the design is the simulation engine, discussed in Section III, which performs the real-time simulation of the system modeled.

This core computes results of the simulated model every time step using the LB-LMC method. Using a dataflow execution approach, the simulation engine computes time step results in a single FPGA clock cycle, this clock cycle period set to the time step of the simulation.

To allow the simulation engine to generate external analog signals, an interface was created that converts result samples from the simulation engine into a form the DAC module can accept. In the DAC interface, DDR-LVDS output converters are used to convert four, 16-bit samples from the simulation engine (normalized from the engine's fixed point) into an interleaved 32-bit DDR-LVDS data bus that is provided to the DAC module to generate analog signals from the samples.

The output DDR-LVDS converters are driven by a clock sourced by the DAC module. The DAC module expects data to be acknowledged with a data clock 90 degrees out of phase with its own clock, so a clock generation (MMCM/PLL) block is used to create the phase shifted clock.

Since the simulation engine and DAC module can be driven by independent clocks which can differ in frequency and phase, asynchronous dual-clock, FIFO buffers are used to pass data without loss between the different clock domains.

If the simulation engine and analog module are driven by the same clock, the FIFO buffers can be forgone, such as done in our HIL platform which uses the clock of the DAC

to drive the simulation engine. Since the simulation engine may need access to digital signals, such as external switch control signals for simulated power converters, general purpose input/output (GPIO) interfaces on the FPGA are exposed to the top-level design and passed to the engine.

For internal converter switching control for the simulation engine, an entity is created that consists of Direct Digital Synthesizer (DDS) blocks to generate sine waveform references and PWM logic to provide open-loop switch control for modeled converters; this internal control is driven by the same clock as the simulation engine.

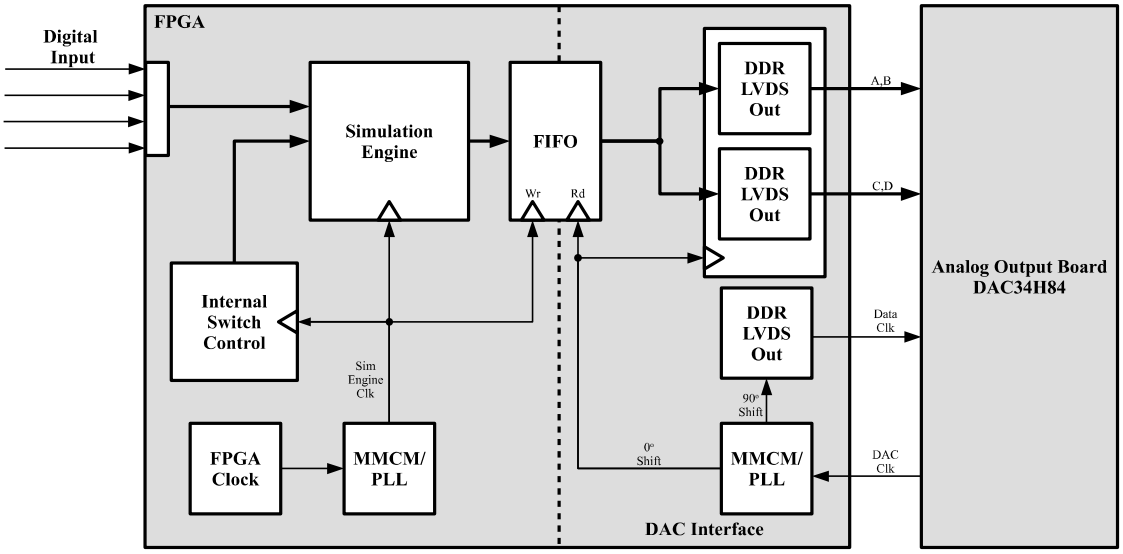


Figure 4.4 FPGA Top-Level Design with DAC Interface

4.4.3 ADC Interfacing

In the case where analog input is to be used in the top-level design for HIL simulation, the design displayed Figure 4.4 is expanded with an interface for receiving converted signals from the ADC module, as seen in Figure 4.5. Similarly, to the DAC interface but in reverse operation, DDR-LVDS input converters are used to convert the interleaved 28-bit data bus from the ADC module into four 14-bit samples that are converted to the fixed-point representation accepted by the simulation engine. The samples

from the ADC module are expected to be registered by the design on every cycle of the clock sourced by the ADC module. If the clock of the ADC module is not same used by the simulation engine, an asynchronous dual-clock FIFO buffer is used between the simulation engine and the ADC interfacing so to ensure that the samples are passed without error from the interface to the simulation engine.

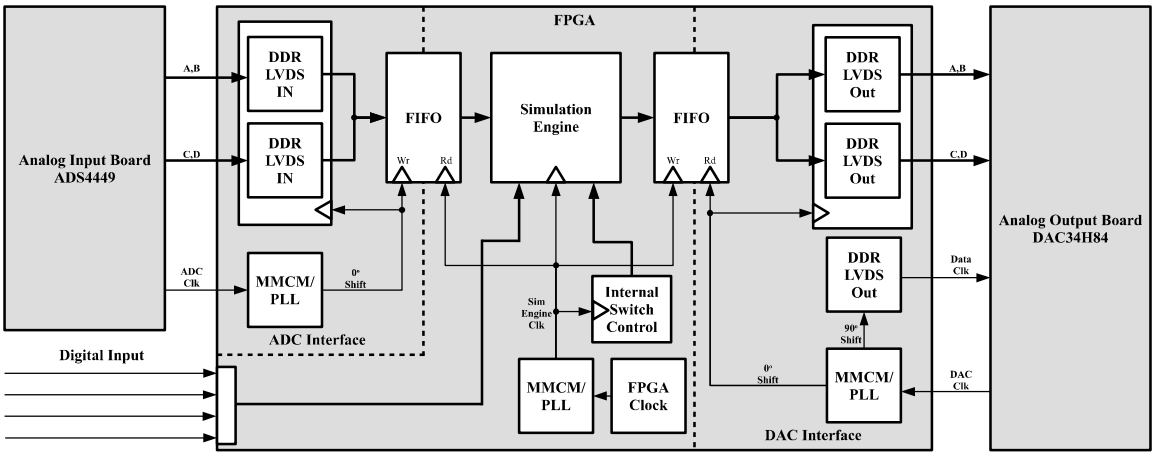


Figure 4.5 FPGA Top-Level Design with Full Analog I/O.

4.4.4 Converter Controller

As explained in Section IV, a single 3-phase converter is configured in the test model to be externally controlled in closed-loop configuration. This converter is controlled by a simple voltage-oriented controller which samples the 3-phase voltages of the converter and produces PWM signals to operate the switches of the converter. A diagram of this controller is shown in Figure 4.6, which is implemented on the Delfino controller mentioned before. This controller begins its update period, set to the desired switching frequency for the converter, by first sampling the analog representation of the 3-phase voltages of the modeled converter simulated on the FPGA, using its own ADC. These voltages are then converted into the DQ domain using Park and Clarke Transformations for effective tracking of the 3-phases to control. Each term of the DQ representation of the

phases is fed into a PI controller where the D term is set to desired phase amplitude by a reference (ref) and Q term is set for zero. The PI controller signals are converted from DQ domain back to 3-phase space and normalized for input to PWM units that generate control signals for the converter's switches.

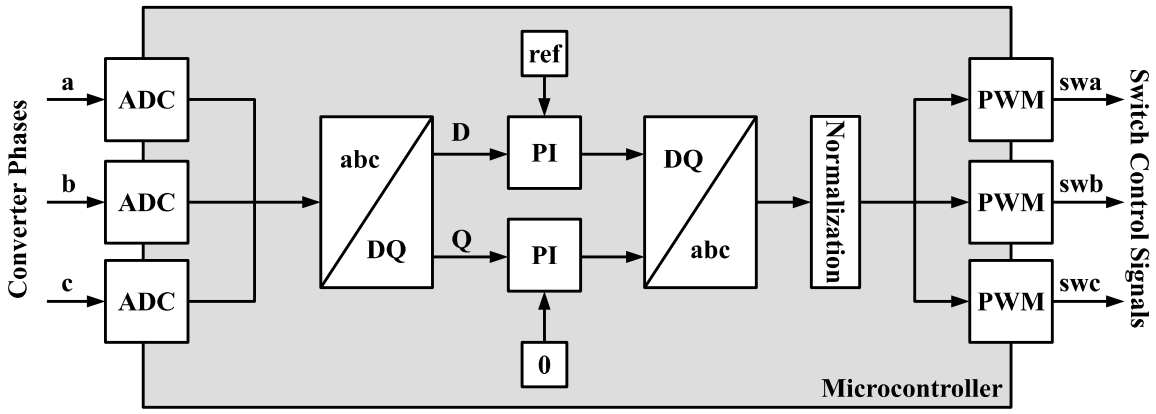


Figure 4.6 Block diagram of the converter's controller.

4.5 Results

In this section, findings from running the RT-HIL simulation platform test bench executing the previously discussed test model and control are presented.

4.5.5 Test Setup

The test bench discussed in Section V was assembled and configured to HIL testing of the notional system discussed in Section IV.

The top-level FPGA design for the simulation engine shown in Figure 4.4 was implemented using Xilinx Vivado HLx 2016.1 suite, with simulation engine developed in C++ and the top-level design in VHDL.

Fixed-point precision was set for 43 bits fractional, 29 bits integral. The controller shown in Figure 8 was implemented using Texas Instruments Code Composer Studio in C++. All converters in the system model were operated with 100kHz switching frequency,

including the externally controlled one. The model was simulated in real-time with 50 nanosecond time step.

4.5.6 Closed Loop Control

While the HIL simulation was executing, modeling the shipboard system to run with 100kHz switching frequency and 50ns time step, the analog signals of the externally controlled converter's three phases were captured via an oscilloscope. During the capturing, the controller was set to change from maximum output amplitude (6kV) to half output amplitude (3kV). The real-time capture of the converter output during the control change is shown Figure. 4.7, the results scaled from the analog output's converted 3.3V range to the model's range of 12kV.

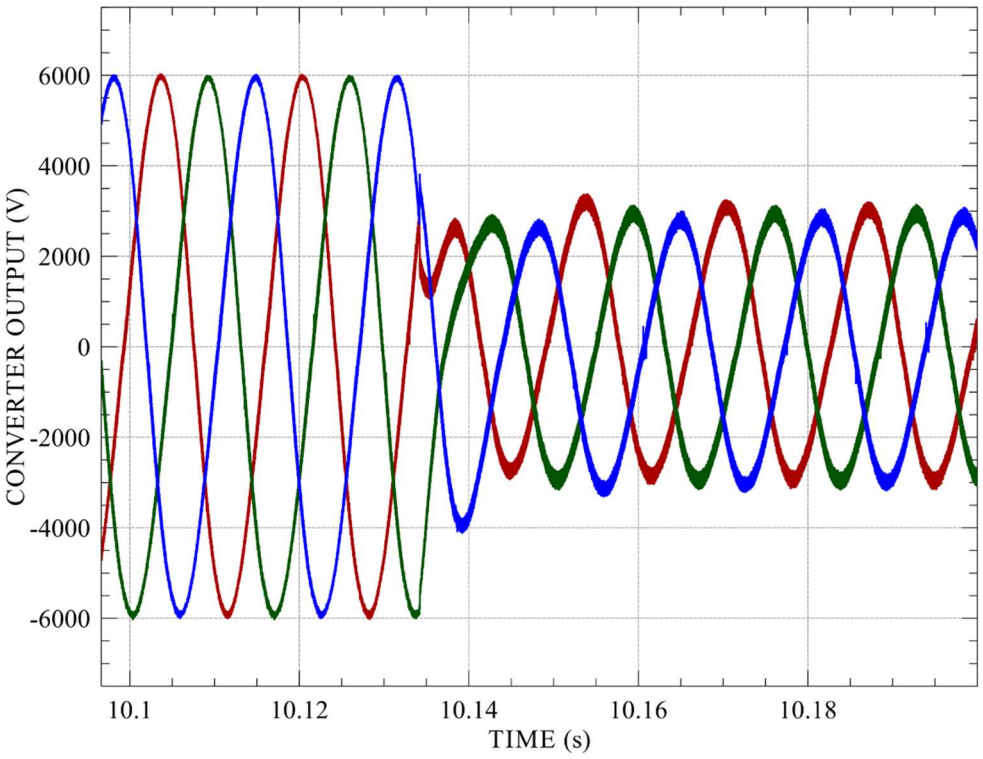


Figure. 4.7 Converter Output after Control Change

Along with control change results, the 100kHz switching ripple found in the converter output was also captured in real-time during another execution of the HIL

simulation while control reference was held constant. Captured result is shown in Figure. 4.8.

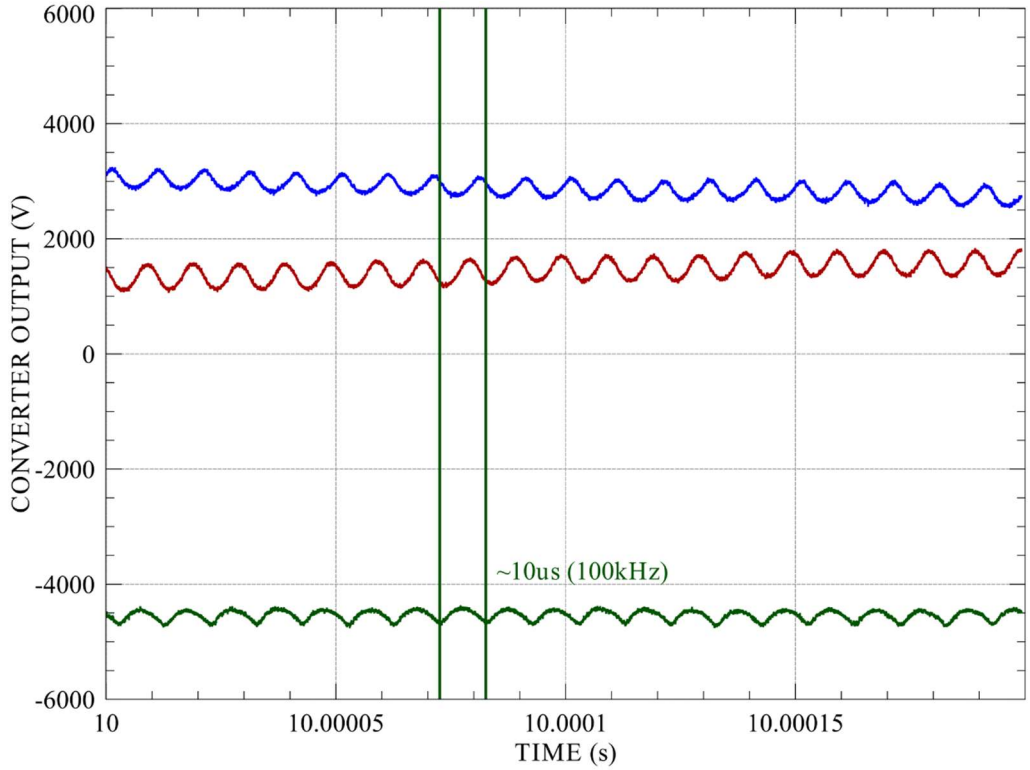


Figure. 4.8 Converter Output Ripple.

4.6 Conclusions

This work presents the implementation of a high-speed, FPGA-based, Hardware in the Loop simulation platform with both digital and analog I/O. This test bench is successfully capable of HIL-simulating large power systems at 50 nanosecond time steps while modeling switching power converters externally controlled in closed-loop at 100kHz switching frequency. While design and development of controls is outside the scope of the proposed dissertation work, a closed loop voltage control unit has been developed and employed to realize the HIL method presented in this.

There are some limitations of the used approach for PEPDS. One main limitation of the presented RT-HIL method regards the lack of multi-FPGA decoupling capabilities

for of simulation of large PEPDS exceeding the current FPGA resource availability. Another limitation regards the interface with the control which is based on analog connections and with a limited number of signals hard wired to the control while some commercial simulator, less performant than the present one, rely on flexible protocols like Aurora which with only one fiber optic cable are capable of transmitting several signals in framing or streaming mode. The lack of this kind of capabilities, poses limitations in connecting the platform to other simulators for co-simulation tests so negatively affecting the flexibility of the simulator.

CHAPTER 5

SYSTEM LEVEL TESTING OF PEPDS

5.1 Introduction

To overcome limitations of state-of-the-art RT and HIL simulation platforms -as illustrated in the literature review of chapter 1-, in this dissertation work various RT-HIL simulation methods have been presented.

In this chapter, so to increase the flexibility of the previously proposed methods, it is introduced a co-simulation method for which a dedicated platform has been implemented. Such platform is based on a multi-CPU commercial test bench - OPAL-RT OP5607- and a custom US+ FPGA based platform-, interacting through the communication interfaces described in chapter 3, running high performances RT simulations based on the LB-LMC method and compatible with converter models developed using the methodology proposed in chapter 2.

This work has contributed to test and develop the Ortis library [56] with which it is compatible and thanks to which the simulation model used for the work presented here has been developed.

This research work contributes to the US Navy “All Electric Ship” project and the main goal of the co-simulation method is to perform accurate system level simulations of the SZS a peculiar PEPDS replicating possible power systems layouts for future electric ships.

5.2 Simulation method, requirements, and limitations

The simulation method employed for this work is based on the co-simulation concept. To optimize the computational resources of the simulation platforms involved, the simulation model is split according to its internal dynamics. The slower part -a PGM- is simulated on the CPU based platform while the fast-switching converter-based portion is simulated on the custom FPGA platform. Due to the different dynamics of the simulated subsystems, an ITM described and analyzed in [57] and illustrated in Figure. 5.1 is employed for decoupling.

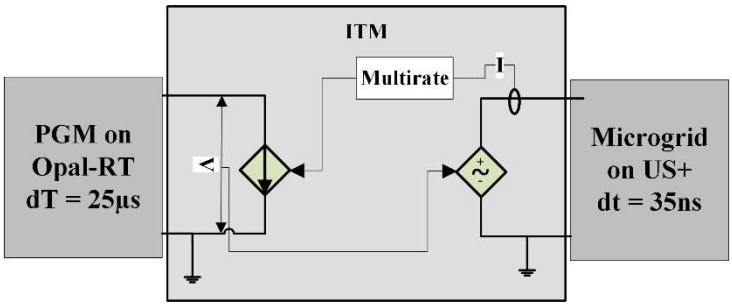


Figure. 5.1 The ITM model used for the co-simulator platform.

As in [58], the ITM employed for the present case decouples subsystems over a DC line specifically the SZS main bus at 12kV and the two decoupled subsystems are a PGM running on Opal at 25µs and a microgrid model based on DAB converters simulated on US+ at 35ns.

A main difference from the previous work consists in the presence of the multi-rate interface.

The faster subsystem requires a multi-rate smoothing unit, like the one presented in chapter 3 to reduce loss of information. The fact the PGM has a very slow dynamic if compared with the microgrid on the multi-FPGA platform justifies the employment of the

ITM which unit step delay introduced in the simulation has negligible impact on the results accuracy. For this reason, the Aurora interface can be applied since the serialization process introduces an acceptable communication latency -falling within the simulation time step of Opal.

The simulation method proposed here has general application for all those PEPDS presenting a clear subdivision between subsystems dynamics. The commercial multi-CPU unit -in this case Opal-RT- facilitates the user interaction with the whole co-simulation platform which especially for the FPGA part can turn to be no user-friendly.

Limitations of this method are mostly related to the requirements of the computational resource limits of the FPGA and the fact the multi-CPU unit as per definition is slower than the FPGA system and in this specific case cannot run RT-simulations below 20μs.

5.3 Simulation model description

In this section it is described the model employed to test the capabilities of the proposed co-simulation method.

The SZS model presented here and illustrated in Figure. 5.2 can be mapped for similarities on models such as the ones described in [58].

The present model consists in a PGM characterized by slow dynamic -hence simulated with a large time step of 25μs- feeding a microgrid characterized by a faster dynamic simulated with a small-time step equal to 35ns.

The model has been decoupled by introducing an ITM on its main DC bus line. This subdivision is coherent with the fact that the PGM being an electro-mechanical system, so characterized by “slow” dynamics will run on the CPU based OPAL-RT unit

while the microgrid constituted by zones each one containing fast switching converters models has a faster dynamic and will run on the custom FPGA side of the co-simulator which being faster than the CPU unit will allow higher accuracy.

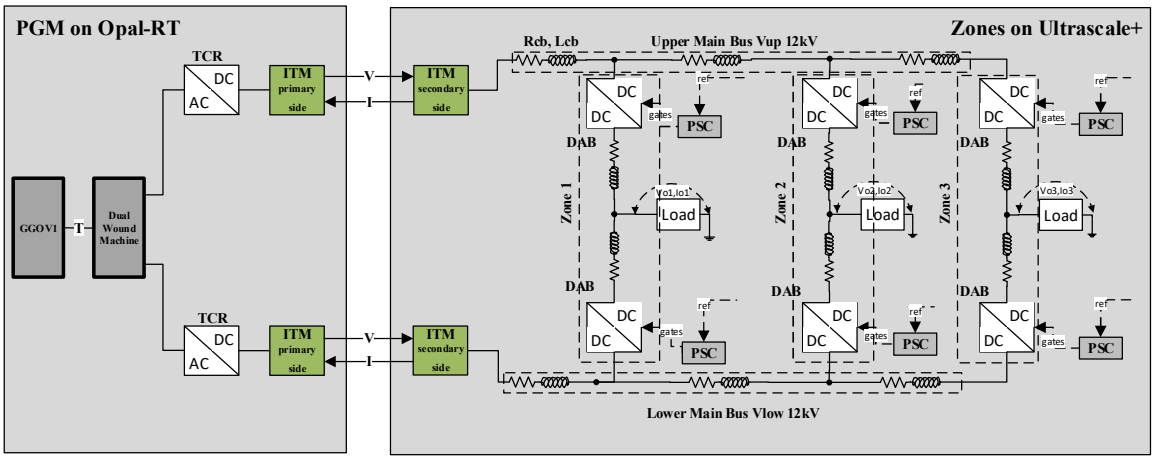


Figure. 5.2 The SZS Model representing the plant of the multi-FPGA co-simulator.

The PGM employs thyristor-based rectifiers to feed the microgrid main DC bus and it is constituted by an AC generator which converts the mechanical power produced by the gas turbine GGOV1 in electrical power through the DWM which is a synchronous machine providing two three phase outputs each one feeding one of the rectifiers.

The gas turbine model follows the standard GGOV1 structure described in [59][60]. It generates as output the active torque Tg and receives as input the rotational speed ω which is used by the gas turbine speed governor for feedback control.

The GGOV1 model embeds a control and a plant. The first one, is an isochronous control of the gas turbine shaft speed ω based on a given set point in per units set to 1.

This is specifically realized through three PI logics controlling the shaft speed, acceleration, and combustion chamber temperature.

The output of each PI in terms of fuel demand is provided to a minimum value selector forwarding its output to the plant.

The microgrid consists in three zones each one containing two DAB converters, feeding a load. Each DAB converter receives gates signals by its dedicated PSC unit.

The SZS model rating is indicated in Table 5.1. The PGM generates 30MW AC. Each rectifier absorbs 15MW and feeds the main bus of the microgrid with 12kV DC. Within each zone of the microgrid, each DAB converter, converts the main bus voltage in 1kV DC to feed the current controlled load.

Table 5.1 PGM rating parameters.

Description	Value
Main PGM power rating	30 MW
Generator frequency	120 Hz
Rectifiers power rating (each)	15 MW
PGM output voltage (DC)	12 kV
DAB load voltage	1kV

5.4 Simulator platform description

A schematic of the proposed co-simulation platform is shown in Figure. 5.3.

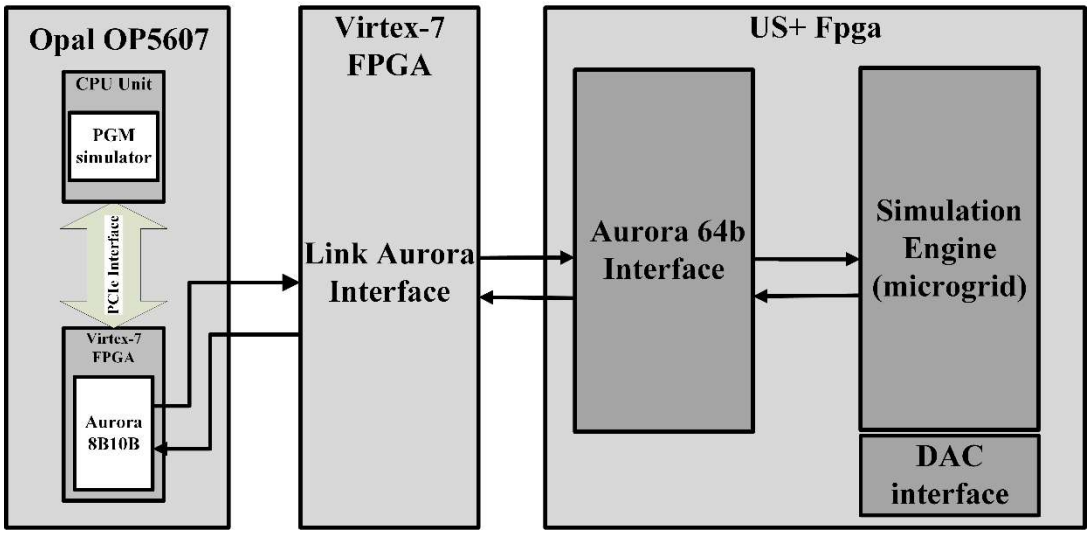


Figure. 5.3 Block diagram of the co-simulation platform design.

The PGM model running on the multi-CPU unit of Opal, communicates with the custom US+ FPGA based simulation through a dedicated communication interface based

on the Aurora protocol, namely “Link Aurora Interface” deeply described in Chapter 3. The multi-CPU unit is a Linux RedHat OS based machine which communicates, through PCIe, with an internal Virtex-7 FPGA where a communication interface based on a hardcoded Aurora 8b is deployed. This design is supported by SFP fiber optics.

On US+, the Aurora 64b interface allows to the SZS microgrid model contained in the simulation engine to communicate with the PGM running on Opal, from which it is decoupled with an ITM.

Results can be monitored and logged on Opal through a proprietary asynchronous data log interface at 25µs communicating with the US+ devices through the Link-Aurora design running on Virtex-7 FPGA or can be monitored and logged in RT on commercial oscilloscopes thanks to a dedicated DDR based DAC interface developed for this work and meant for for TI-DAC devices.

5.5 Laboratory setup

The laboratory setup employed to perform the RT co-simulation according to the proposed simulation method previously described is illustrated in Figure. 5.4.

The simulator hardware is composed of a commercial part based on the OPAL-RT OP5607 test bench and a custom part based on different Xilinx Virtex family FPGAs. The OP5607 is more suitable for simulating slow and complex dynamic systems models. It allows deploying models on its CPUs unit via MATLAB Simulink while the FPGA platform is more suitable to host fast dynamic switching converter-based power systems due to its parallel computation capabilities.

The user’s access point to the simulation platform is a host-PC which can be any commercial CPU based machine compatible with the RT-LAB software which allows to

handle the Opal test bench. From here the user can visualize the ongoing simulation waveforms, log the results, and interact with the simulation through a console interface.

The Virtex-7 FPGA running the Link Aurora interface is connected to all the other devices through fiber optics cables and FMC connector on which a FMC to SFP adapter card, specifically the FM-S14 card providing four SFP+ ports, is plugged on.

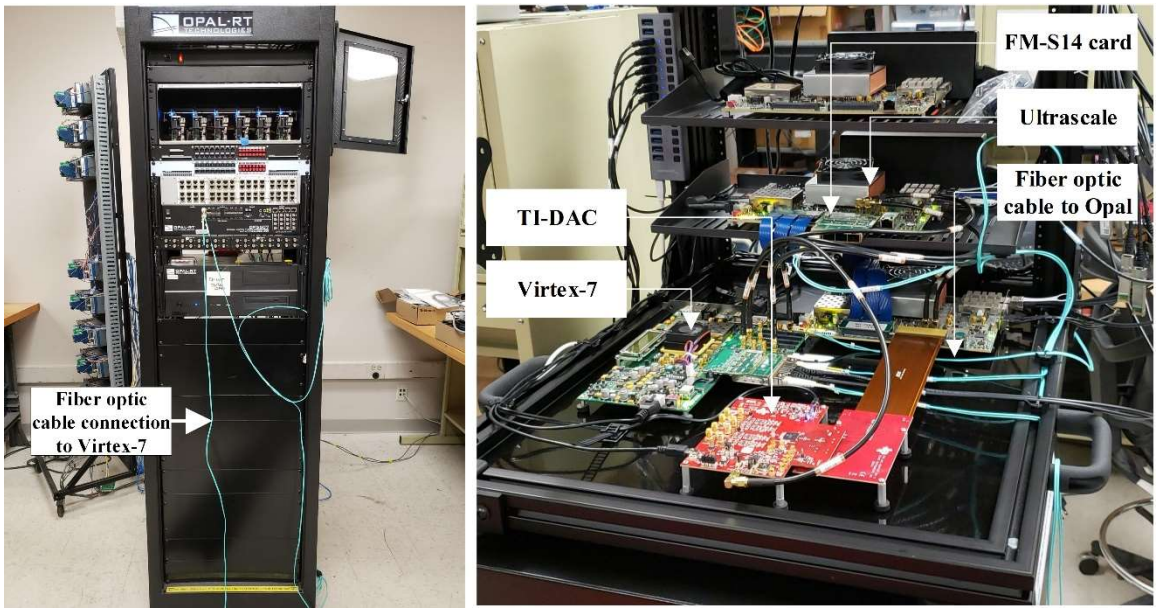


Figure. 5.4 The laboratory setup employed to perform the RT co-simulation.

The US+ FPGA is connected to the oscilloscope through a TI-DAC device plugged on one of its FMC connectors through a flexible brown FMC cable.

5.6 Experimental results

In this section are reported the experimental results of a RT co-simulation experiment conducted on the co-simulation platform presented in the previous sections.

To demonstrate the capabilities of the Aurora based communication interface developed to realize the co-simulation, the results presented in Figure 5.5 have been logged employing the asynchronous Opal data log interface at 25µs.

In the first plot it is illustrated the transient of the gas turbine rotor speed which starting from zero per units (pu) reaches its regime after 100sec. At this point the exciter of the DWM kicks-in providing the necessary electro-magnetic field to the DWM rotor and the user from the RT-Lab console can interact with the simulation activating the microgrid zones running at 35ns on the US+ FPGA. In this condition all the DAB converters within each zone are activated and their primary (blue waveform) and secondary (green waveform) internal transformers voltages starts toggling following the same behavior of the DAB RT simulation presented in Chapters 2 and 3.

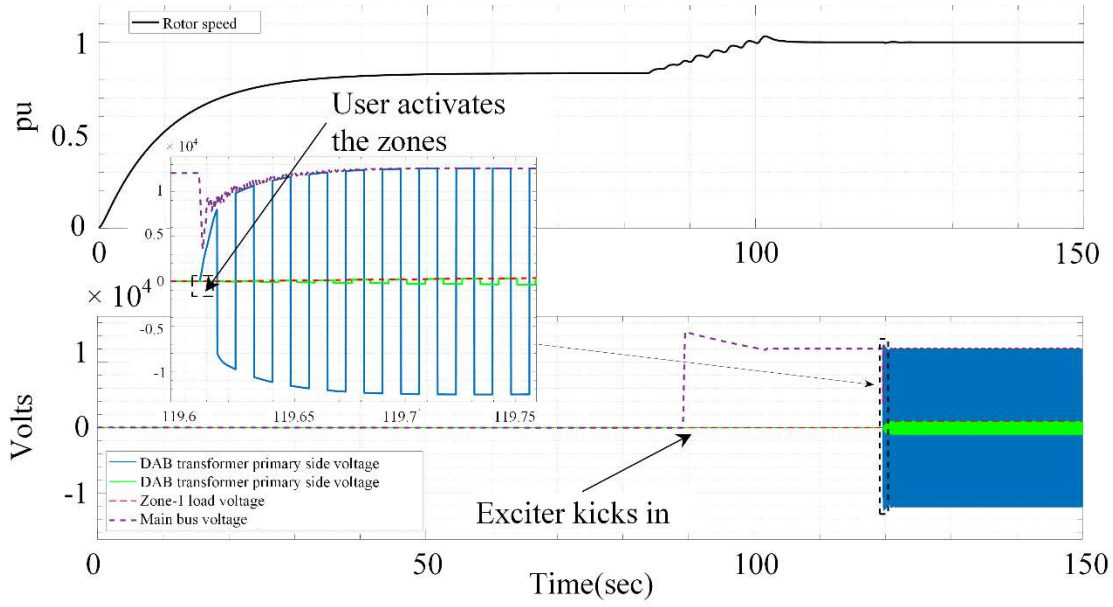


Figure 5.5 Results from the RT co-simulation of the SZS model.

5.7 Conclusions

In this chapter, so to provide an additional component to increase the flexibility of the RT-HIL simulation methods proposed in this dissertation, a co-simulation method based on Aurora protocol is introduced. The method aims to simulate at system level large PEPDS keeping a high level of accuracy in a scalable and flexible way. For this purpose, a co-simulation platform has been developed.

Such platform involves an OPAL-RT multi-CPU unit and a custom FPGA based platform. Specifically, the co-simulation platform is based on a communication interface supported by the Aurora protocol.

The target of the simulation is a complex PEPDS model for electric ship applications which has been deployed and simulated on the platform where the slow electro-mechanical part has been simulated on Opal at 25µs while the fast-switching converters portion part -based on DAB converters- has been simulated in single rate with a fixed discrete simulation time step 35ns on US+ FPGA applying the LB-LMC method.

The results demonstrate how it is possible to simulate a complex and large size microgrid with different dynamics at system level. The co-simulation helps to optimize the computational resources by dividing the system in subsystems, which according to their dynamics are deployed on a CPU or FPGA platform. Due to the slow dynamic of the system simulated in RT on OPAL, the decoupling realized with an ITM has negligible effects on the accuracy. The system implementation is facilitated by Simulink and ORTiS. The employment of the OPAL-RT platform facilitates the implementation of complex systems with Simulink. The ORTiS tool helps realizing a full C++ microgrid model just by providing it a netlist of the various fast switching converters components.

The developed RT-HIL methodology for the presented work has a general application for PEPDS and the SZS model simulated for this works was just an example to show the simulator capabilities.

Regarding future developments, the application of the LB-LMC nodal decomposition method can extend the simulation of the microgrid solver over multiple FPGAs connected with a fast communication interface such as the parallel bus. This would

allow to simulate a large system with fast switching converters at 70ns with possibility to log results in RT through a dedicated TI-DAC device. Furthermore, the developed Aurora interface for co-simulation can be adapted and employed to realize CHIL experiments with external control units communicating with the microgrid model running on the multi-FPGA platform which performances might be further enhanced focusing future development work on the parallel bus FSM simulation execution and communication time steps which currently are equal to 35ns. Future FPGA boards with increased capabilities may allow to reduce the FPGA design routing delay typical of large microgrid models so to allow further improvements on simulation results accuracy.

.

CHAPTER 6

CONCLUSIONS

The main achievement of this dissertation research work consists in developing a RT and HIL methods to conduct system level tests on PEPDS with high level of accuracy in a scalable and flexible way.

The target system modelled is the US Navy SZS which is an electromechanical power system including a gas turbine, a DWM and a microgrid based on DAB converters in DC/DC configuration grouped in zones.

The developed RT simulation platforms includes a multi-CPU unit based on OPAL-RT technologies where the slower dynamics portion of the SZS is simulated at 25μs and a custom multi-FPGA simulation platform, based on Virtex US+, which can simulate the pure electrical portions of the SZS using small simulation time steps equal to 35ns on mono FPGA and 70ns for multi-FPGA configurations employing the parallel bus interface.

Some important contributions of this work consist in the realization of state space full switching models of MMC rectifiers and DAB converters, implementation of the SZS electro-mechanical model portion, realization of communication interfaces to allow communication between OPAL-RT and the custom FPGA based platform and implementation of a fast parallel bus interface to realize the nodal decomposition of PEPDS models on the multi-FPGA platform. The developed RT simulation methods shows how high level of accuracy for system level tests of PEPDS models can be achieved in a scalable

and flexible way. The flexibility introduced by Aurora based interfaces simplify the co-simulation and Child connections while the LB-LMC nodal decomposition method supported by the newly developed parallel bus interface extends the scalability of the LB-LMC solver from a single to multiple FPGAs. The realization of the simulation, in terms of model implementation is facilitated by the ORTiS tool.

Regarding future developments of this research work, the RT-HIL simulation methods including the co-simulation method presented in the final chapter, might be integrated with each other leading to even more increased RT-HIL simulation performances. Also, the parallel bus interface might be further improved with future FPGA devices which with proper clocking and design routing capabilities may lead to a reduction in the simulation time step below 70ns. Further extension of this interface beyond three FPGA devices might be realized by improving the hardware IO capabilities of the device where the system manager design runs

REFERENCES

[1] M. Difronzo, M. Milton, M. Davidson, and A. Benigni, "Hardware-In-The-loop testing of high switching frequency power electronics converters," *2017 IEEE Electr. Sh. Technol. Symp. ESTS 2017*, no. 1, pp. 299–304, 2017.

[2] Q. Deng, X. Liu, R. Soman, M. Steurer, and R. A. Dougal, "Primary and backup protection for fault current limited MVDC shipboard power systems," *2015 IEEE Electr. Sh. Technol. Symp. ESTS 2015*, pp. 40–47, 2015.

[3] E. De Din, G. Lipari, M. Cupelli, and A. Monti, "Hardware-in-the-loop validation of control parameters for MVDC shipboard power systems," *2016 Int. Conf. Electr. Syst. Aircraft, Railw. Sh. Propuls. Road Veh. Int. Transp. Electrif. Conf. ESARS-ITEC 2016*, 2016.

[4] S. Ziaeinejad and A. Mehrizi-Sani, "Software-based hardware-in-the-loop real-time simulation of distribution systems," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2016-Novem, 2016.

[5] G. Yang *et al.*, "A three-level boost-buck converter for the ultracapacitor applications," *PEDG 2019 - 2019 IEEE 10th Int. Symp. Power Electron. Distrib. Gener. Syst.*, pp. 700–704, 2019.

[6] Xilinx.Inc, "Aurora 8B/10B Protocol Specification."

[7] Xilinx.Inc, "Aurora 64B/66B Protocol Specification.

[8] Z. Shen, T. Duan, and V. Dinavahi, "Design and Implementation of Real-Time Mpsoc-FPGA-Based Electromagnetic Transient Emulator of CIGRÉ DC Grid for HIL Application," *IEEE Power Energy Technol. Syst. J.*, vol. 5, no. 3, pp. 104–116, 2018.

[9] T. Duan, Z. Shen, and V. Dinavahi, "Multi-Rate Mixed-Solver for Real-Time Nonlinear Electromagnetic Transient Emulation of AC/DC Networks on FPGA-MPSoC Architecture," *IEEE Power Energy Technol. Syst. J.*, vol. 6, no. 4, pp. 183–194, 2019.

[10] X. Ma *et al.*, "Full-scale physical control system real-time digital simulator design for MMC-HVDC simulation and testing," *Proc. IECON 2017 - 43rd Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 2017-Janua, pp. 4518–4523, 2017.

[11] N. Lin and V. Dinavahi, "Behavioral Device-Level Modeling of Modular Multilevel Converters in Real Time for Variable-Speed Drive Applications," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 5, no. 3, pp. 1177–1191, 2017.

[12] N. Lin and V. Dinavahi, "Dynamic Electro-Magnetic-Thermal Modeling of MMC-Based DC-DC Converter for Real-Time Simulation of MTDC Grid," *IEEE Trans. Power Deliv.*, vol. 33, no. 3, pp. 1337–1347, 2018.

[13] H. Saad, S. Member, T. Ould-bachir, J. Mahseredjian, and C. Dufour, "Real-Time Simulation of MMCs Using CPU and FPGA," *IEEE Trans. Power Electron.*, vol. 30, no. 1, pp. 259–267, 2015.

[14] X. Zhai *et al.*, "Multi-rate real-time simulation of modular multilevel converter for HVDC grids application," *Proc. IECON 2017 - 43rd Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 2017-Janua, pp. 1325–1330, 2017.

[15] M. Rivard, C. Fallaha, A. Yamane, J. Paquin, M. Hicar, and C. J. P. Lavoie, "Real-Time Simulation of a More Electric Aircraft Using a multi-FPGA Architecture," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 5760–5765.

[16] W. Li, L. A. Gregoire, P. Y. Robert, S. Souvanlasy, and J. Belanger, "Modular Multilevel Converter model implemented in FPGA for HIL test of industrial controllers," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2014-Octob, no. October, pp. 0–4, 2014.

[17] W. Li, L. A. Grégoire, S. Souvanlasy, and J. Bélanger, "An FPGA-based real-time simulator for HIL testing of modular multilevel converter controller," *2014 IEEE Energy Convers. Congr. Expo. ECCE 2014*, pp. 2088–2094, 2014.

[18] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems," *IET Gener. Transm. Distrib.*, vol. 7, no. 5, pp. 451–463, 2013.

[19] T. Liang and V. Dinavahi, "Real-time device-level simulation of MMC-based MVDC traction power system on MPSoC," *IEEE Trans. Transp. Electrif.*, vol. 4, no. 2, pp. 626–641, 2018.

[20] Z. Li *et al.*, "Power Module Capacitor Voltage Balancing Method for a ±350-kV/1000-MW Modular Multilevel Converter," *IEEE Trans. Power Electron.*, vol. 31, no. 6, pp. 3977–3984, 2016.

[21] R. Mo, M. Steurer, and H. Li, "Controller hardware-in-the-loop (CHIL) simulation of a multi-functional energy storage system based on modular multilevel DC/DC converter (M2DC) for MVDC grid," *2016 IEEE 8th Int. Power Electron. Motion Control Conf. IPEMC-ECCE Asia 2016*, pp. 1980–1984, 2016.

[22] M. Matar, D. Paradis, and R. Iravani, “Real-time simulation of modular multilevel converters for controller hardware-in-the-loop testing,” *IET Power Electron.*, vol. 9, no. 1, pp. 42–50, 2016.

[23] A. Benigni and A. Monti, “A parallel approach to real-time simulation of power electronics systems,” *IEEE Trans. Power Electron.*, vol. 30, no. 9, pp. 5192–5206, 2015.

[24] M. Milton, A. Benigni, and A. Monti, “Real-Time Multi-FPGA Simulation of Energy Conversion Systems,” *IEEE Trans. Energy Convers.*, vol. 34, no. 4, pp. 1–1, 2019.

[25] J. Wang *et al.*, “State-space switching model of modular multilevel converters,” *2013 IEEE 14th Work. Control Model. Power Electron. COMPEL 2013*, 2013.

[26] J. Wang, R. Burgos, and D. Boroyevich, “Switching-cycle state-space modeling and control of the modular multilevel converter,” *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 2, no. 4, pp. 1159–1170, 2014.

[27] G. J. Kish and P. W. Lehn, “Modeling Techniques for Dynamic and Steady-State Analysis of Modular Multilevel DC/DC Converters,” *IEEE Trans. Power Deliv.*, vol. 31, no. 6, pp. 2502–2510, 2016.

[28] M. Milton, “LB-LMC Solver C++ Code Generation Library,” 2019. [Online]. Available: https://github.com/MatthewMilton/LBLMC-CodeGen.

[29] M. M. Biswas and M. Z. R. Khan, “Amended THD with modified phase-shifted PWM for micro-grid connected multilevel inverter,” *2017 IEEE Power Energy Conf. Illinois, PECI 2017*, no. 1, 2017.

[30] R. Rodrigues, J. Li, and H. L. Ginn, “A novel frequency domain control method for modular multilevel converters under non-sinusoidal supply conditions,” *2017 IEEE Energy Convers. Congr. Expo. ECCE 2017*, vol. 2017-Janua, pp. 1506–1512, 2017.

[31] F. Zhang, G. Joos, W. Li, and J. Belanger, “A validation methodology for real time model of modular multilevel converter,” in *2015 IEEE Energy Conversion Congress and Exposition, ECCE 2015*, 2015, pp. 3344–3350.

[32] A. Zama, A. Benchaib, S. Bacha, D. Frey, and S. Silvant, “High Dynamics Control for MMC Based on Exact Discrete-Time Model With Experimental Validation,” *IEEE Trans. Power Deliv.*, vol. 33, no. 1, pp. 477–488, 2018.

[33] S. Ji, X. Huang, J. Palmer, F. Wang, and L. M. Tolbert, “Modular Multilevel Converter (MMC) Modeling Considering Submodule Voltage Sensor Noise,” *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 1215–1219, 2021.

[34] H. Qin and J. W. Kimball, “Generalized average modeling of dual active bridge DC-DC converter,” *IEEE Trans. Power Electron.*, vol. 27, no. 4, pp. 2078–2084, 2012.

[35] K. Zhang, Z. Shan, and J. Jatskevich, “Large- and Small-Signal Average-Value Modeling of Dual-Active-Bridge DC-DC Converter Considering Power Losses,” *IEEE Trans. Power Electron.*, vol. 32, no. 3, pp. 1964–1974, 2017.

[36] R. Mallik *et al.*, “Equivalent Circuit Models of Voltage-controlled Dual Active Bridge Converters,” *2019 IEEE 20th Work. Control Model. Power Electron. COMPEL 2019*, pp. 1–4, 2019. [37] B. Singh, G. Shankar, and A. Singh, “Modelling of inverter interfaced dual active bridge converter,” *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 680–685, 2019.

[38] H. Shi, K. Sun, H. Wu, and Y. Li, “A Unified State-Space Modeling Method for a Phase-Shift Controlled Bidirectional Dual-Active Half-Bridge Converter,” *IEEE Trans. Power Electron.*, vol. 35, no. 3, pp. 3254–3265, 2020.

[39] H. Shi, K. Sun, H. Wu, Y. Li, and X. Xiao, “Unified state-space modeling method for dual-active-bridge converters considering bidirectional phase shift,” *2018 IEEE Energy Convers. Congr. Expo. ECCE 2018*, pp. 643–649, 2018.

[40] A. S. Description, “Full Discrete-Time Modeling and Stability Analysis of the Digital Controlled Dual Active Bridge Converter,” no. 51207126, 2016.

[41] S. Gao, X. Lu, and X. Liu, “Minimum reflux power control of bidirectional DCDC converter based on dual phase shifting,” *2018 IEEE 2nd Int. Electr. Energy Conf.*, no. 51337001, pp. 440–445, 2018.

[42] Xilinx.Inc, *High-Speed Serial I/O Made Simple*. Xilinx, 2005.

[43] Xilinx, “Xilinx, ‘Aurora 8B/10B Protocol Specification,’ SP002 v2.3, Oct. 2014.”

[44] A. Benigni, A. Monti, and R. A. Dougal, “Latency-based approach to the simulation of large power electronics systems,” *IEEE Trans. Power Electron.*, vol. 29, no. 6, pp. 3201–3213, 2014.

[45] M. Milton, A. Benigni, and J. Bakos, “System-Level, FPGA-Based, Real-Time Simulation of Ship Power Systems,” vol. 32, no. 2, pp. 737–747, 2017.

[46] M. Difronzo, M. Milton, M. Davidson, and A. Benigni, “Hardware-In-The-loop testing of high switching frequency power electronics converters,” in *2017 IEEE Electric Ship Technologies Symposium, ESTS 2017*, 2017.

[47] W. Chen, T. Liang, and V. Dinavahi, “Comprehensive Real-Time Hardware-In-the-Loop Transient Emulation of MVDC Power Distribution System on Nuclear Submarine,” *IEEE Open J. Ind. Electron. Soc.*, vol. 1, no. November, pp. 326–339, 2020.

[48] F. R. Goldstein, “Bus-oriented switching system for asynchronous transfer mode,” 5452330, 1995.

[49] IBM, “32-bit Processor Local Bus, Architecture Specifications.” .

[50] “PCI LOCAL BUS, ‘PCI Local Bus Specification’, Revision 2.2.” 1998.

[51] M. Milton and A. Benigni, “Software and Synthesis Development Libraries for Power Electronic System Real-Time Simulation,” *2019 IEEE Electric Ship Technologies Symposium, ESTS 2019*. pp. 368–376, 2019.

[52] M. Milton and A. Benigni, “ORTiS solver codegen: C++ code generation tools for high performance, FPGA-based, real-time simulation of power electronic systems,” *SoftwareX*, vol. 13, p. 100660, 2021.

[53] J. Liu and V. Dinavahi, “A real-time nonlinear hysteretic power transformer transient model on FPGA,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3587–3597, 2014.

[54] R. E. Crosbie, J. J. Zenor, D. Word, R. Bednar, and N. G. Hingorani, “A low-cost high-speed real-time simulator for ships power systems,” *2011 IEEE Electr. Sh. Technol. Symp. ESTS 2011*, pp. 102–105, 2011.

[55] R. Meka, M. Sloderbeck, O. Faruque, J. Langston, and M. Steurer, “FPGA model of a High-frequency Power Electronic Converter in an RTDS Power System Co-simulation,” pp. 71–75, 2013.

[56] M. Milton and A. Benigni, “Ortis.” [Online]. Available: https://github.com/OpenRealTimeSimulation.

[57] M. Stevic, A. Monti, and A. Benigni, “Development of a simulator-to-simulator interface for geographically distributed simulation of power systems in real time,” *IECON 2015 - 41st Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 5020–5025, 2015.

[58] O. N. R. G. N---, “Model Description Document Notional Four Zone MVDC Shipboard Power System Model,” 2019.

[59] M. R. Bank Tavakoli, B. Vahidi, and W. Gawlik, “An educational guide to extract the parameters of heavy duty gas turbines model in dynamic studies based on operational data,” *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1366–1374, 2009.

[60] L. W. Rowen, “Simplified mathematic representations of heavy-duty gas turbines,” *J. Eng. Gas Turbines Power*, 1983.