Electric Power Systems Research 141 (2016) 147–156

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

An automated FPGA real-time simulator for power electronics and power systems electromagnetic transient applications

R. Razzaghi*, M. Mitjans, F. Rachidi, M. Paolone

École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 23 February 2016
Received in revised form 10 June 2016
Accepted 15 July 2016

Keywords:
Real-time simulation
Electromagnetic transients
FPGA
Hardware-in-the-loop

ABSTRACT

The paper presents an automated FPGA-based real-time electromagnetic transients simulator for power electronics and power systems applications. The simulator features an automated procedure enabling its straightforward applications to different topologies by avoiding the need of complex FPGA programming. The proposed solver integrates: (i) the Modified Augmented Nodal Analysis (MANA) method, (ii) the Fixed Admittance Matrix Nodal Method (FAMNM), (iii) the optimal selection of the switch conductance parameter, and (iv) efficient sparse matrix-to-vector multiplier. The simulator is able to accurately reproduce, in real-time, electromagnetic transients taking place in power electronics devices together with electromagnetic waves propagating in transmission lines. The peculiar structure of the MANA-FAMNM solver enables to reach extremely low integration time steps and avoids the need to redesign the FPGA code. The results of the proposed simulator are validated by dedicated comparisons with off-line EMTP-RV simulations and a hardware-in-the-loop (HIL) test for a three-phase two-level inverter and a three-phase power network.

1. Introduction

Electromagnetic transient (EMT) real-time simulators (RTSs) are essential tools in power systems and power electronics to design, analyze and test control processes [1], and to understand dynamics of complex systems and specific devices. RTSs have been evolving from the so-called Transient Network Analyzer (TNA) to digital RTSs [1]. This last type can make use of hardware based on multi-core CPUs, general-purpose processors (GPPS), digital signal processors (DSPs) or computer clusters. In general, CPU based RTSs represent a better option to simulate bulk power networks since they can achieve acceptable simulation time steps (e.g., in the order of few tens of microseconds) and model relatively complex systems. Additionally, existing CPU-based RTSs are typically linked to the well-established programming environments (e.g., MATLAB SimPowerSystems (SPS)). However, the achievable integration time steps of CPU-based RTSs have a lower bound associated with the partial sequential operations that the CPU architectures need to deploy. As a consequence, the relatively large simulation time steps required by these simulators do not allow to model high frequency phenomena such as EMTs in power converters or travelling wave transients taking place in transmission lines (e.g., faults and switching transients).

During the past years, the size and computational power of FPGAs has been increased dramatically. FPGA-based real-time simulation has emerged as a leading trend for HIL applications for the power electronics applications (e.g., Refs. [2–5]). The parallel processing hardwired in FPGAs enables the implementation of specific methodologies that dramatically reduce the sequencing of the operations taking place in CPUs. FPGA-RTSs provide lower sampling rate, higher frequency bandwidth and lower I/O latency [6]. These characteristics position this type of simulators as better candidates for the accurate real-time simulation of power electronics and limited-size power systems. Indeed, one of the main drawbacks of the FPGA-RTSs is their difficult programming associated with the use of Hardware Description Languages (HDL). This low level programming limits the scalability of solvers and, as a consequence, the representation of complex models. Additionally, any modification in the implemented algorithm/model requires, in general, a time-consuming recompilation of the HDL code. Indeed, long offline validation procedure, together with the long code compiling time, complicates the development of sophisticated circuit models and controls deployable in this specific hardware platform. In

* Corresponding author. Tel.: +41 21 693 48 15; fax: +41 21 693 46 62.
E-mail addresses: reza.razzaghi@epfl.ch, r.razzaghi87@gmail.com (R. Razzaghi).

http://dx.doi.org/10.1016/j.epsr.2016.07.022
0378-7796/

this respect, the automation of FPGA-RTSs is a challenging research topic that has been discussed, for instance, in Ref. [7].

With particular reference to EMT-RTS applications, the real-time simulation of electrical circuits requires a topological mapping of both propagative (transmission lines) and lumped elements of the circuit. Their number and interconnection is not known a-priori. Therefore, the construction of the problem equations (see Section 3) is case-dependent. As a consequence, the number of tasks involving FPGA resources such as memories, FIFOs, DSP block, etc. is not known a-priori, making the generalization of the code difficult. The developed FPGA-RTS should be capable of simulating a given network without the need for the a-priori knowledge about that. Of course, such a simulation is deployable within the allocation limits of the FPGA resources.

Another challenging issue regarding this type of simulators is the limitation in the matrix manipulation and the inherent difficulty to represent switching maneuvers. In this respect, the most straightforward method to represent topology-variable circuits in FPGA-RTS is the so-called Fixed Admittance Matrix Nodal Method (FAMNM) [3]. This method, irrespective of the number of the switches, allows for obtaining a time-invariant nodal admittance matrix during switching transitions. However, it introduces artificial oscillations and errors in the simulation results (e.g., Ref. [8]).

It is worth mentioning that, few automated FPGA-RTSs have been proposed to avoid the difficulties of the FPGAs programming (e.g., Refs. [9,10]). However, the applications of these simulators are mainly dedicated to the power electronics HIL simulations. Therefore, the above works have not accounted for the possibility to simulate power networks which include propagative transmission lines. Furthermore, these industrial RTSs do not take into account the optimal tuning of the discrete-time switch conductance values to minimize the errors in the simulation results [8].

Within this context, and to address the above listed challenges, this paper proposes an automated FPGA-RTS for both power electronics and power systems applications. In particular, this paper extends the work presented in Ref. [11] by presenting: (i) a complete description of the proposed solver integrating the FAMNM and propagative transmission line models, (ii) the definition of the automated FPGA-RTS and the coupling of the solver with existing off-line simulators like the EMTP-RV simulation environment, (iii) the integration into the solver of the method proposed in Ref. [8] to find the optimal value of the conductance in discrete-time switch model, (iv) the deployment of an efficient sparse matrix-to-vector multiplier, and, (v) the validation of the proposed simulator with respect to power electronics and power systems case studies using off-line simulations and experimental results obtained by means of dedicated HIL test setup.

Compared to the existing literature in the context of FPGA-RTS for the power electronics and power systems applications (e.g., Refs. [2–4,12,13]) this paper proposes (i) an automated procedure to define the FPGA solver parameters, (ii) high performance FPGA-RTS thanks to the parallel implementation of the solver and use of efficient matrix-to-vector multiplier, (iii) systematic validation by means of quantitative error assessments using both offline simulation results and measurements obtained from an HIL test, and (iv) accurate simulation results thanks to the optimal selection of the switch parameter.

The structure of the paper is as follows. Section 2 provides a brief overview of the EMT simulation. Section 3 describes the hardware and the proposed algorithm for the developed FPGA real-time simulator. Section 4 illustrates the FPGA solver architecture. Section 5 presents the validation of the proposed simulator. Section 6 concludes the paper with final remarks.

2. Circuit analysis and models adopted in the proposed FPGA-RTS

2.1. Circuit analysis and numerical integration methods

As known, two main types of solution methods are used in circuit analysis: (i) nodal and (ii) state-space ones [1]. Within the context of FPGA-based real-time simulation, nodal analysis (or Modified Augmented Nodal Analysis—MANA) is preferable since it allows straightforward formulation of the system equations and, in particular, it enables the FAMNM approach. We briefly recall the MANA formulation [14]:

$$[x]_{n\times 1} = [A]^{-1}_{n\times n} \times [J]_{n\times 1} \tag{1}$$

where matrix $[A]$ is formed by the discrete representation of the network elements; $[x]$ is the vector of unknowns including the network's node voltages and branch currents; and $[J]$ is a vector of the independent sources and current history terms related to the network components. For EMT simulation applications, trapezoidal and backward-Euler methods are the most popular numerical integration methods [15]. In this paper, this latter has been adopted.

2.2. Discrete models of network components

1) Lumped elements (R, L, C)

The most common approach for discrete-time representation of the network elements is the one proposed in Ref. [16] where the circuit elements are converted into their Norton equivalent. The lumped elements (L, C) are represented by an equivalent conductance in parallel with a current source representing the history values [14,16]. The values for the equivalent conductance and the history current source are determined by the element type (i.e., R, L, C) together with the adopted numerical integration method [16].

2) Transmission lines

The so-called Bergeron model [16,17] allows a straightforward representation of constant-parameter transmission line models. As known, this approach is based on a circuit representation of the telegraphers' equations where each line termination is replaced by means of a lumped impedance in parallel with a controlled current source. By using the modal analysis, the same approach can be extended to the case of multi-conductor lines, in which the multi-conductor equations are decoupled by using phase-to-modal transformation matrices (e.g. Refs. [16,18]). The use of Bergeron model can be also adopted to represent frequency-dependent line models [19] even if in our solver we limit the analysis to constant-parameter line models.

3) Switches

For FPGA real-time simulations, the use of detailed switch models is prohibitively time consuming. Therefore, behavioral switch models have been proposed [20]. Among them, the simplest ones are the ideal switch model or the so-called two-valued resistor model where two different resistors are associated with each state of the switch (R_{off}, R_{on}). However, for such models, the system's admittance matrix needs to be updated and re-factorized after each switching change, generating major issues to satisfy the FPGA computational time constraints. On the contrary, the use of the discrete-time switch model allows defining the so-called FAMNM. In this case, the switch is represented by a relatively small inductance when its state is 'closed' and by a relatively small capacitance when its state is 'open' (e.g., Refs. [21,22]). As a consequence, the