

Physics-Based Device-Level Power Electronic Circuit Hardware Emulation on FPGA

Wentao Wang, *Student Member, IEEE*, Zhuoxuan Shen, *Student Member, IEEE*,
and Venkata Dinavahi, *Senior Member, IEEE*

Abstract—Accurate models of power electronic devices are necessary for hardware-in-the-loop (HIL) simulators. This paper proposes a digital hardware emulation of device-level models for the insulated gate bipolar transistor (IGBT) and the power diode on the field programmable gate array (FPGA). The hardware emulation utilizes detailed physics-based nonlinear models for these devices, and features a fully paralleled implementation using an accurate floating-point data representation in VHSIC hardware description language (VHDL) language. A dc–dc buck converter circuit is emulated to validate the hardware IGBT and diode models, and the nonlinear circuit simulation process. The captured oscilloscope results demonstrate high accuracy of the emulator in comparison to the offline simulation of the original system using Saber software.

Index Terms—Field-programmable gate arrays (FPGAs), hardware-in-the-loop (HIL), insulated gate bipolar transistor (IGBT), parallel algorithms, parallel processing, physics-based modeling, power diode, power electronic circuit simulation.

I. INTRODUCTION

HARDWARE-IN-THE-LOOP (HIL) simulators are prevalent [1]–[6] in many industrial applications, and field programmable gate arrays (FPGAs) are playing a significant role in their success due to their fast processing times and advanced toolset for rapid prototyping of new concepts and designs. For power electronic system modeling in HIL, simulators simpler models such as the ideal switching function model or the averaged model are heretofore employed. Admittedly, such models are sufficient for system-level performance evaluation and analysis, e.g., to observe system-level waveforms and harmonic analysis. Furthermore, the simulation process is typically a fixed time-step, piecewise linear, and noniterative method. Device-level models for power electronic switches have not seen application in HIL simulators mainly due to their computational burden. Nevertheless, high-fidelity device-level modeling does have its benefits, especially due to the current trend in miniaturization of power electronics in various applications whose performance can be influenced by high-frequency switching devices. HIL simulators will be increasingly required to model such systems to develop novel

Manuscript received June 13, 2014; accepted September 26, 2014. Date of publication October 07, 2014; date of current version November 04, 2014. This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC). Paper no. TII-14-0894.

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: wentao@ualberta.ca; zshen@ualberta.ca; dinavahi@ualberta.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2014.2361656

solutions. Detailed device-level models would provide insight into high-frequency effects such as device stresses, parasitics, electromagnetic interference (EMI) noise issues, and their suppression. Potential areas of application could also include statistical, parametric, and sensitivity analyses for power electronic system design. Even if no closed-loop design evaluation is required, achieving fast execution will be a significant milestone to accelerate device-level modeling and simulation, e.g., it takes several minutes to simulate even a few milliseconds of circuit behavior using device-level models in currently available offline software-based simulators such as Saber and PSpice. An accurate physics-based power electronic device digital hardware emulation has not yet been reported in literature, although approximate device-level real-time simulation has been reported: using linearized device-level characteristics of the insulated gate bipolar transistor (IGBT) [7], and using look-up tables [8] to model the nonlinear switching characteristics of the IGBT. The main challenges of emulating device-level models include: 1) solution of typically a large set of coupled nonlinear equations; 2) ability to do this within a time-step of a few nanoseconds to capture the high-frequency device behavior; 3) to perform the emulation using a variable time-step for computational efficiency; and 4) to model enough nonlinear switches in terms of circuit size for practical application. We attempt to do this on the FPGA architecture, due to its intrinsic advantages of full hardware parallelism, and user reconfigurability. Due to the advances in very large scale integration (VLSI) technology and in digital hardware design tools, FPGAs are finding widespread application [9]–[17]. Newer generations of FPGAs have much larger capacity in terms of logic blocks, distributed memories, DSP slices, and other advanced features such as partial reconfiguration that enable complex model and algorithm emulation.

IGBTs and power diodes are the fundamental switching elements in modern power electronic systems. They also contribute significantly to the circuit nonlinearity. Much literature has been devoted to device-level model development for both the IGBT and the power diode. Device-level models for these two types of switches can be mainly classified into: 1) physics-based analytical models; 2) behavioral models; and 3) numerical models. As the name implies, the physics-based models describe the carrier dynamics in the semiconductor material in terms of nonlinear ordinary or partial differential equations with respect to time. The numerical solution methodology normally includes implicit discretization followed by Newton or Katzenelson iterations. For the IGBT, two of the most detailed and popular models in this category are the

Hefner [18], [19], and Kraus [20] models, and for the power diode the lumped charge models proposed by Lauritzen and Ma [21], [22] are quite famous. The behavioral models (macromodels) [24], [25] approximate the device behavior from measured and fitted characteristics that are incorporated into the simulation using lookup tables and discrete circuit components. Bond graph methods to build averaged models for power converters that include device nonlinearities have been proposed in [26]. The numerical models are highly exact two-dimensional (2-D) or three-dimensional (3-D) models [27] based on nonlinear partial differential equations in space-time that describe physical phenomena in the semiconductor material such as carrier generation, recombination, drift, and diffusion. The numerical solution methodology typically involves the finite element or the finite difference methods. To reduce high computational burden of such methods, hybrid models [28] have also been proposed which combine the analytic and numerical models.

This paper presents a digital hardware emulation of the IGBT and the power diode physics-based device-level models on the FPGA. The device nonlinear equations are solved using a fully iterative Newton–Raphson method with a variable time-step. The IGBT hardware emulation utilizes the physics-based Hefner’s model, and the power diode hardware emulation utilizes the Lauritzen’s model. The proposed FPGA-based emulation also includes other power electronic system elements such as independent/dependent supply sources, linear/nonlinear lumped R-L-C elements, and pulsewidth modulation, which makes it a complete power electronic circuit emulator. A variable time-step implicit discretization of the device equations is employed to gain computational advantage, and result comparison is presented with respect to Saber. The entire hardware design is fully paralleled and deeply pipelined using IEEE 32-bit floating-point number precision to achieve high speed and accuracy requirements. The paper is organized as follows. Section II describes the physics-based IGBT and power diode model hardware emulation on the FPGA. The overall hardware framework for the power electronic circuit emulator and the simulation algorithms are given in Section III. Case studies and experimental results are presented in Section IV, and conclusion in Section V.

II. PHYSICS-BASED DEVICE MODEL HARDWARE EMULATION

A. Power Diode Module

1) *Model Formulation*: In contrast to the normal p–n diode, the p–i–n structure has a wide intrinsic region between the p-layer and n-layer, which enables it to work in fast switching and high-voltage operations, standard for most power diodes [29]. Other type of diodes such as the Schottky diode only work in specific scenarios [30]. Therefore, for power converter hardware emulation, a detailed model for p–i–n power diode is desirable. The physics-based p–i–n diode model contains the following five major phenomenological characteristics: 1) reverse recovery; 2) forward recovery; 3) emitter recombination; 4) junction capacitance; and 5) contact resistance

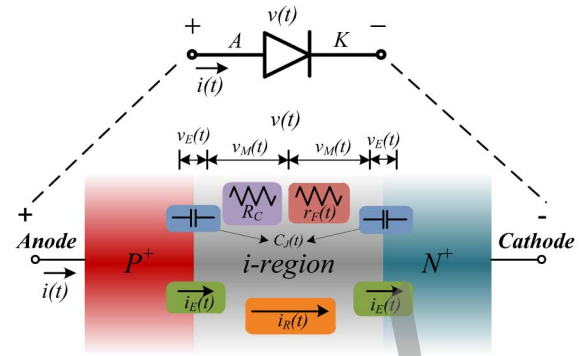


Fig. 1. Physical structure of power p–i–n diode.

(see Fig. 1). This model is based on the lumped charge concept [21], [22], and utilizes transport equations in the semiconductor to represent the diode physics.

The reverse recovery phenomenon is represented by the following equations:

$$i_R(t) = \frac{q_E(t) - q_M(t)}{T_M} \quad (1)$$

$$0 = \frac{dq_M(t)}{dt} + \frac{q_M(t)}{\tau} - \frac{q_E(t) - q_M(t)}{T_M} \quad (2)$$

$$q_E(t) = I_S \tau [e^{\frac{v_E(t)}{V_T}} - 1] \quad (3)$$

where $i_R(t)$ is the diffusion current from the center of the intrinsic region; $v_E(t)$ is the junction voltage; $q_E(t)$ is the junction charge variable; $q_M(t)$ is the charge in the middle of intrinsic region; T_M is the diffusion transit time; τ is the carrier lifetime; I_S is the diode saturation current constant; and the constant V_T is the thermal voltage. The forward recovery phenomenon is described by

$$v_M(t) = \frac{V_T T_M R_{M0} i(t)}{q_M(t) R_{M0} + V_T T_M} \quad (4)$$

where $i(t)$ represents the whole diode current, $v_M(t)$ is voltage across half of the intrinsic region, and R_{M0} is the initial resistance in the intrinsic region. The emitter recombination phenomenon is formulated as

$$i_E(t) = I_{SE} [e^{\frac{2v_E(t)}{V_T}} - 1] \quad (5)$$

where $i_E(t)$ is the end region recombination current, and I_{SE} is the emitter saturation current constant. The contact resistance R_C is modeled by

$$v(t) = 2v_M(t) + 2v_E(t) + R_C \cdot i(t) \quad (6)$$

where $v(t)$ is the voltage across the diode. Finally, the charge stored in the junction capacitance $q_J(t)$ which contributes to the diode current $i(t)$ is described by

$$i(t) = i_R(t) + i_E(t) + \frac{dq_J(t)}{dt}. \quad (7)$$

The junction capacitance $C_J(t)$ and its charge $q_J(t)$ are determined as

$$C_J(t) = \begin{cases} \frac{C_{J0}}{[1 - \frac{2v_E(t)}{V_J}]^m}, & v_E < \frac{V_J}{4} \\ \frac{m \cdot C_{J0} \cdot 2v_E(t)}{V_J \cdot 0.5^{m+1}} - (m-1) \frac{C_{J0}}{0.5}, & v_E \geq \frac{V_J}{4} \end{cases} \quad (8)$$

$$q_J(t) = \int C_J(t) d[2v_E(t)] \quad (9)$$

where C_{J0} is the zero-biased junction capacitance, V_J is the junction built-in potential, and m is the junction grading coefficient.

2) *Model Discretization and Linearization:* As seen from above, the power diode model described by (1)–(9) is nonlinear and time-varying. Here, we discretize and linearize the model to obtain a discrete-time equivalent circuit for time-domain simulation. The charge $q_M(t)$ differential equation (2) is first discretized using the Trapezoidal rule as

$$q_M(t) = \frac{\Delta t}{2T_M(1 + \frac{k_1 \Delta t}{2})} \cdot q_E(t) + \frac{q_{\text{hist}}(t - \Delta t)}{1 + \frac{k_1 \Delta t}{2}} \quad (10)$$

where the history charge is given as

$$q_{\text{hist}}(t) = q_M(t) \left(1 - \frac{\Delta t}{2} \cdot k_1\right) + \frac{\Delta t}{2T_M} \cdot q_E(t) \quad (11)$$

and

$$k_1 = \frac{1}{\tau} + \frac{1}{T_M}. \quad (12)$$

Similarly, the differential term $\frac{dq_J(t)}{dt}$ [represented by $i_J(t)$, the current contributed by junction capacitor] in (7) is discretized as

$$i_J(t) = \frac{2}{\Delta t} \cdot q_J(t) - \frac{2}{\Delta t} \cdot q_J(t - \Delta t) - i_J(t - \Delta t) \quad (13)$$

where the junction capacitor charge $q_J(t)$ can be deduced from

$$q_J(t) = \begin{cases} \frac{-V_J C_{J0}}{1-m} \cdot \left[1 - \frac{2v_E(t)}{V_J}\right]^{(1-m)}, & v_E(t) < \frac{V_J}{4} \\ \frac{2^{m+2} m C_{J0} v_E^2(t)}{V_J} - 2^{m+1} (m-1) C_{J0} v_E(t), & v_E(t) \geq \frac{V_J}{4}. \end{cases} \quad (14)$$

Meanwhile, the nonlinearity of the power diode model is obvious from (3), (4), (5), and (8). The nonlinearity coming from the reverse recovery characteristics (3) is linearized as a dynamic conductance g_R and a parallel current source i_{Req} given as

$$g_R = \frac{\partial i_R}{\partial(2v_E)} = k_2 I_{ST} \cdot \frac{1}{2V_T} \cdot e^{\frac{v_E(t)}{V_T}} \quad (15)$$

$$i_{\text{Req}} = k_2 I_{ST} \left[e^{\frac{v_E(t)}{V_T}} - 1 \right] - k_3 q_{\text{hist}}(t - \Delta t) - g_R \cdot 2v_E(t) \quad (16)$$

where

$$k_2 = \frac{1}{T_M} - \frac{\Delta t}{2T_M^2(1 + \frac{\Delta t k_1}{2})} \text{ and } k_3 = \frac{1}{T_M(1 + \frac{\Delta t k_1}{2})}. \quad (17)$$

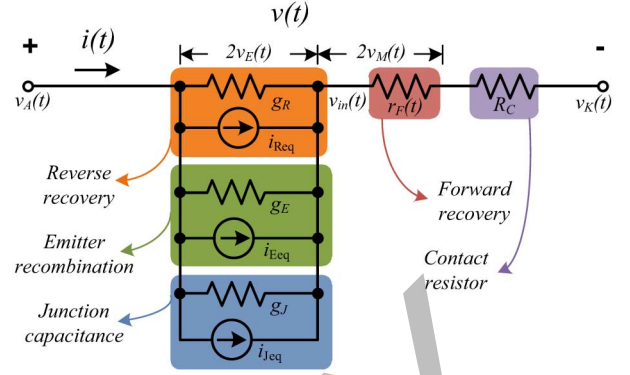


Fig. 2. Discrete-time linearized equivalent circuit for the power diode.

Similarly, the emitter recombination (5) and the junction capacitance (8) nonlinearities are linearized to obtain the corresponding conductance and current source pairs (g_E, i_{Eq}) and (g_J, i_{Jeq}) , respectively

$$g_E = \frac{\partial i_E}{\partial(2v_E)} = \frac{I_{SE}}{V_T} e^{\frac{2v_E(t)}{V_T}} \quad (18)$$

$$i_{\text{Eq}} = I_{SE} \left[e^{\frac{2v_E(t)}{V_T}} - 1 \right] - g_E \cdot 2v_E(t) \quad (19)$$

$$g_J = \frac{\partial i_J}{\partial(2v_E)} = \frac{2}{\Delta t} C_J(t) \quad (20)$$

and

$$i_{\text{Jeq}} = i_J(t) - g_J \cdot 2v_E(t). \quad (21)$$

The forward recovery effect (4) is represented as a time-varying resistor

$$r_F(t) = \frac{2v_M(t)}{i(t)} = \frac{2V_T T_M R_{M0}}{q_M(t) R_{M0} + V_T T_M}. \quad (22)$$

Finally, the complete discretized- and linearized-power diode (Fig. 2) can be written in the form

$$\mathbf{G}^{\text{Diode}} \cdot \mathbf{v}^{\text{Diode}} = \mathbf{i}_{\text{eq}}^{\text{Diode}} \quad (23)$$

where $\mathbf{G}^{\text{Diode}}$ is a 3×3 conductance matrix given as

$$\begin{pmatrix} g_R + g_E + g_J & -g_R - g_E - g_J & 0 \\ -g_R - g_E - g_J & g_R + g_E + g_J + \frac{1}{r_F(t) + R_C} & -\frac{1}{r_F(t) + R_C} \\ 0 & -\frac{1}{r_F(t) + R_C} & \frac{1}{r_F(t) + R_C} \end{pmatrix}$$

and $\mathbf{v}^{\text{Diode}} = [v_A \ v_{\text{in}} \ v_K]^T$ is the node voltage vector and $\mathbf{i}_{\text{eq}}^{\text{Diode}} = [-i_{\text{Req}} - i_{\text{Eq}} - i_{\text{Jeq}} \ i_{\text{Req}} + i_{\text{Eq}} + i_{\text{Jeq}} \ 0]^T$ is the equivalent current source vector.

3) *Hardware Emulation on FPGA:* The power diode hardware module includes six units, which execute in three stages (Fig. 3). The junction limit unit runs in Stage 1, which is used to limit the p–n junction voltage value during the Newton iterations. Specifically, the input signal v_{E_old} (history v_E value from the previous Newton iteration) is updated by the output signal v_{E_new} within the current iteration for this unit. This limit models a bounded p–n junction voltage for the diode exponential characteristic. Otherwise, the values related to the diode

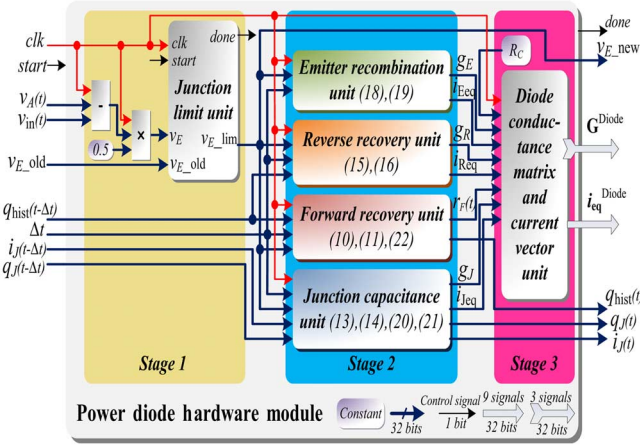


Fig. 3. Architecture of power diode hardware module.

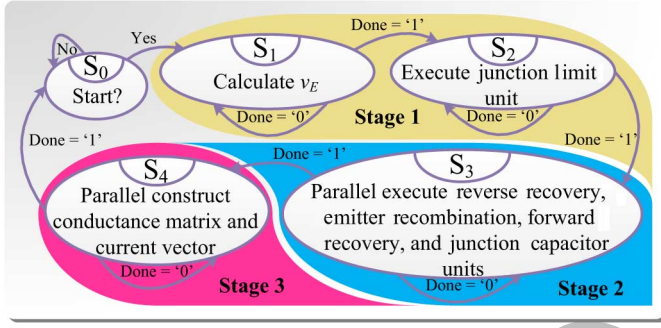


Fig. 4. Finite state machine for the power diode module.

module will be so high that the system equations may become singular and could not be solved, and sometimes they could even hardly be represented as 32-bit floating point numbers. Stage 2 comprises of four units executed in parallel: the emitter recombination unit, the reverse recovery unit, the forward recovery unit, and the junction capacitance unit. Corresponding dynamic conductances and equivalent current sources calculated by these four units are used to form the conductance matrix and the equivalent current source vector in Stage 3. The sequence of operations for this module is also reflected in the finite state machine (Fig. 4), where all the four aforementioned units execute simultaneously in State S_3 , before the formation of $\mathbf{G}^{\text{Diode}}$ and $i_{\text{eq}}^{\text{Diode}}$ within the next state S_4 .

Furthermore, the hardware structure of each of the six units is highly paralleled. For example, within the reverse recovery unit (Fig. 5), the calculations of the current i_R and conductance g_R are fully paralleled. To save hardware resources, as can be seen part of the resources are shared and contribute to both calculations. The result is a highly efficient and fast hardware architecture. Terms such as $\frac{k_1 T_M^2}{\tau I_S}$ and $\frac{1}{V_T}$ are precalculated and used as constants in the hardware design, which reduce a great deal of computational complexity and latency. All the basic computation components such as adder, multiplier, and nonlinear operators such as e^x and $1/x$ were generated using the IP Catalog in Xilinx Vivado Design Suite [23].

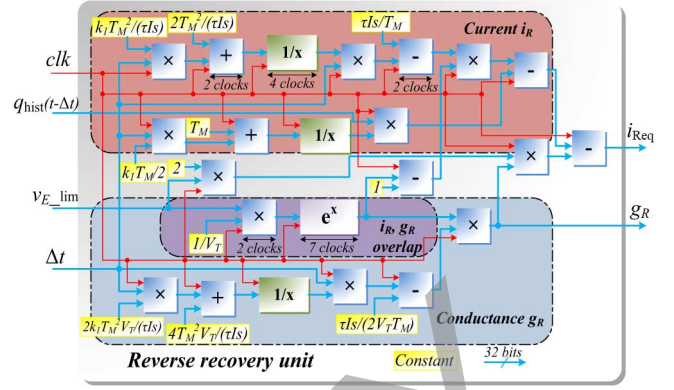


Fig. 5. Hardware structure of the reverse recovery unit.

B. IGBT Module

The IGBT has become the preferred switching device in a great many power electronics applications, because it has both the advantages of fast switching speed and low conduction losses. In order to analyze and estimate a range of device and circuit behaviors (e.g., transients and power loss), an accurate IGBT model is essential. As one of the most detailed-analytical IGBT models, Hefner's physics-based model [18], [19] has been adopted [31] by popular device-level circuit simulators such as Saber and PSpice.

1) *Model Formulation:* According to [18], the IGBT contains the following two main phenomenological characteristics: 1) internal metal oxide semiconductor field effect transistor (MOSFET) phenomena; and 2) internal bipolar junction transistor (BJT) phenomena. The base current of the BJT is fed by the MOSFET, thereby combining the advantages of low ON-state resistance plus high-current capacity of the power BJT and excellent gate drive control of power MOSFET. However, both the internal conceptual BJT and MOSFET function differently from their microelectronic counterparts, because their structures differ greatly based on their design goals. Specifically, Hefner interpreted the intrinsic nonlinear physical phenomena into circuit elements (nonlinear capacitors and dependent current sources, see Fig. 6), which enables device-level circuit simulators to implement the dynamic phenomenological model of IGBT.

The MOSFET channel current is modeled as a current source i_{mos} , which is determined by

$$i_{\text{mos}} = \begin{cases} 0, & v_{\text{gs}} < V_{\text{th}} \\ \frac{K_p K_f [(v_{\text{gs}} - V_{\text{th}}) v_{\text{ds}} - \frac{K_f |v_{\text{ds}}| v_{\text{ds}}]}{1 + \theta (v_{\text{gs}} - V_{\text{th}})}], & |v_{\text{ds}}| \leq \frac{v_{\text{gs}} - V_{\text{th}}}{K_f} \\ \frac{K_p (v_{\text{gs}} - V_{\text{th}})^2}{2[1 + \theta (v_{\text{gs}} - V_{\text{th}})]}, & v_{\text{ds}} > 0 \\ \frac{-K_p (v_{\text{gs}} - V_{\text{th}})^2}{2[1 + \theta (v_{\text{gs}} - V_{\text{th}})]}, & v_{\text{ds}} < 0 \end{cases} \quad (24)$$

where v_{gs} is the gate-source voltage, v_{ds} the drain-source voltage, K_p is transconductance parameter, K_f is triode region transconductance factor, V_{th} is the channel threshold voltage, and θ the transverse field transconductance factor [31].

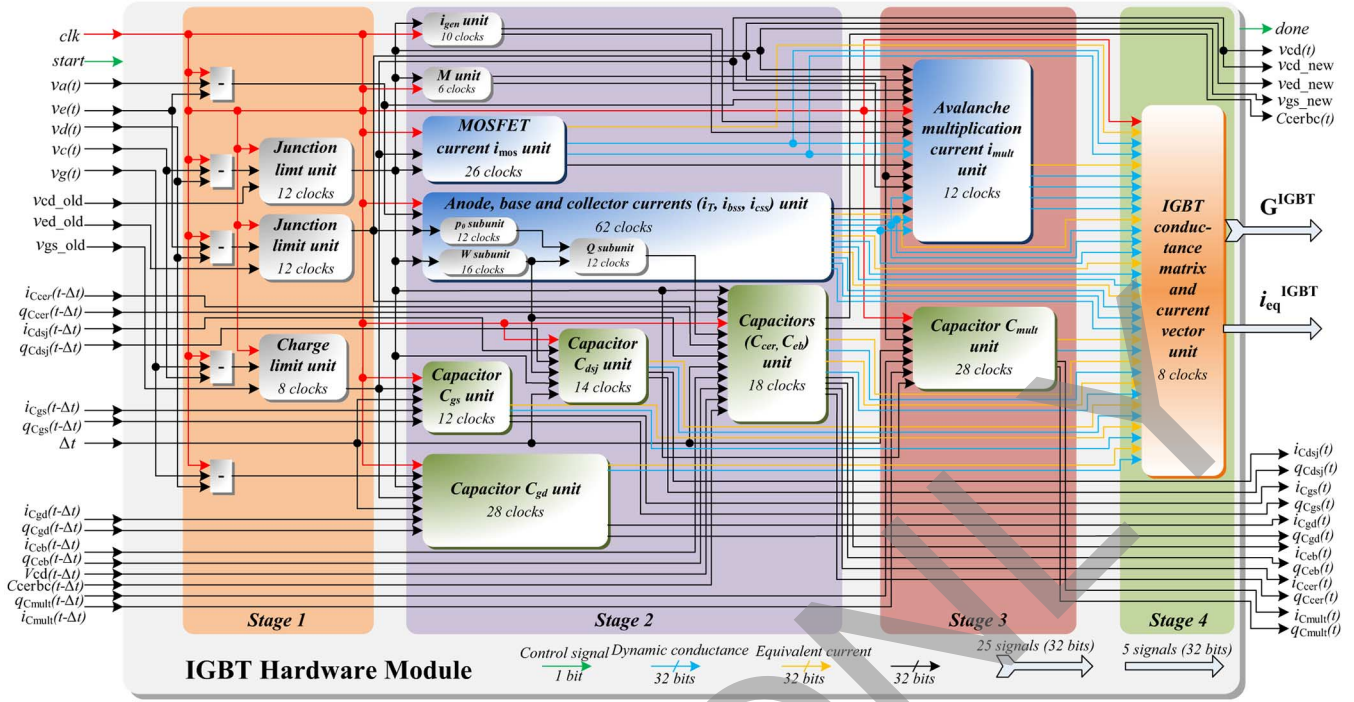


Fig. 8. Architecture of the IGBT hardware module with all subunits horizontally scaled with respect to latency.

current sources (e.g., i_{bsseq}), and 10 voltage-controlled current sources (e.g., $g_{bssbc} \cdot v_{bc}$). All the capacitances (except for C_{cer}) are modeled as pairs of conductance and equivalent source (e.g., g_{Cgs} and i_{Cgsseq}). For example, the drain-source capacitance C_{dsj} has its conductance g_{Cdsj} formulated as

$$g_{Cdsj} = \begin{cases} 0, & W = W_B \\ \frac{2\epsilon_{si}(A - A_{gd})}{\Delta t \cdot (W_B - W)}, & W \neq W_B \end{cases} \quad (34)$$

where A_{gd} is the gate-drain overlap area and the equivalent current source $i_{Cdsjseq}$ is formulated as

$$i_{Cdsjseq} = i_{Cdsj} - g_{Cdsj}v_{ds} \quad (35)$$

where

$$i_{Cdsj} = \frac{2}{\Delta t} [q_{Cdsj} - q_{Cdsj}(t - \Delta t)] - i_{Cdsj}(t - \Delta t) \quad (36)$$

and

$$q_{Cdsj} = qN_B(A - A_{gd})(W_B - W). \quad (37)$$

In contrast, the current sources (e.g., i_{mos}) are modeled as not only pairs of conductance and equivalent source (except for i_{css}), but also as voltage-controlled current sources (e.g., $g_{mosgs} \cdot v_{gs}$). The MOSFET channel current i_{mos} , e.g., has its conductance g_{mosgs} ($\frac{\partial i_{mos}}{\partial v_{gs}}$) formulated as

$$g_{mosgs} = \begin{cases} \frac{K_p K_f v_{ds} - i_{mos} \theta}{1 + \theta(v_{gs} - V_{th})}, & |v_{ds}| \leq \frac{v_{gs} - V_{th}}{K_f} \\ \frac{K_p(v_{gs} - V_{th}) - i_{mos} \theta}{1 + \theta(v_{gs} - V_{th})}, & v_{ds} > 0 \\ -\frac{K_p(v_{gs} - V_{th}) - i_{mos} \theta}{1 + \theta(v_{gs} - V_{th})}, & v_{ds} < 0 \\ G_{min}, & v_{gs} < V_{th} \end{cases} \quad (38)$$

where G_{min} is the MOSFET minimum conductance. The drain-source conductance g_{mosds} ($\frac{\partial i_{mos}}{\partial v_{ds}}$) is formulated as

$$g_{mosds} = \begin{cases} \frac{K_p K_f (v_{gs} - V_{th} - K_f |v_{ds}|)}{1 + \theta(v_{gs} - V_{th})}, & |v_{ds}| \leq \frac{v_{gs} - V_{th}}{K_f} \\ 0, & v_{ds} > 0 \\ 0, & v_{ds} < 0 \\ G_{min}, & v_{gs} < V_{th} \end{cases} \quad (39)$$

and its equivalent current i_{moseq} can be obtained as

$$i_{moseq} = i_{mos} - g_{mosds}v_{ds} - g_{mosgs}v_{gs}. \quad (40)$$

All aforementioned conductances (including those formulated in the voltage-controlled current sources) and equivalent current sources are finally combined to form a 5×5 conductance matrix \mathbf{G}^{IGBT} and current source vector \mathbf{i}_{eq}^{IGBT} , which satisfy the equation

$$\mathbf{G}^{IGBT} \cdot \mathbf{v}^{IGBT} = \mathbf{i}_{eq}^{IGBT} \quad (41)$$

where $\mathbf{v}^{IGBT} = [v_c \ v_g \ v_a \ v_d \ v_e]^T$ is the IGBT node voltage vector, and \mathbf{G}^{IGBT} and \mathbf{i}_{eq}^{IGBT} are given as (47) and (48), respectively, in the Appendix.

3) *Hardware Emulation on FPGA*: The basic strategy of the IGBT model hardware design is to turn all the components of the analog equivalent circuit [Fig. 6(b)] into 14 hardware units (see Fig. 8), to fully exploit the possibility of hardware parallelism. Also, it is highly efficient to code and debug all the hardware units individually. The customizable nature of the IGBT hardware module makes it extremely flexible to fit into different power converter topologies with varying module complexities. All the current units including i_{mos} , (i_T , i_{bss} , and i_{css}), and i_{mult} are responsible for the calculations of the corresponding currents, dynamic conductances, and equivalent

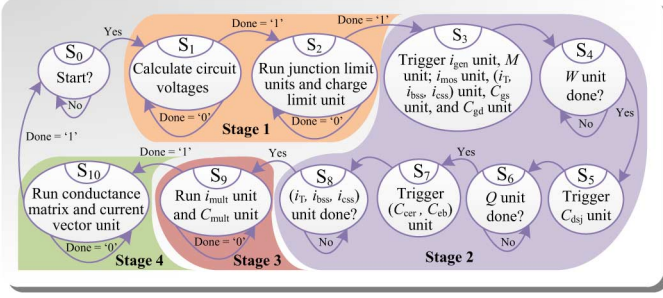


Fig. 9. Finite state machine of the IGBT hardware module.

current sources. All the capacitance units including the C_{gs} , C_{gd} , C_{dsj} , (C_{cer}, C_{eb}) , and C_{mult} are responsible for the calculations of not only the currents, dynamic conductances and equivalent currents, but also the charges. The junction limit units and the conductance matrix and current vector unit have similar functions to those used in the power diode hardware module. Other units such as the charge limit unit is used to limit the charge variation range between successive Newton iterations, while the avalanche factor unit calculates M , and the i_{gen} unit calculates i_{gen} and its base-collector conductance, which are essential for the calculations in the i_{mult} unit and C_{mult} unit. The input signals to the IGBT hardware module include the circuit node voltages such as v_c (cathode/source/collector), v_g (gate), v_a (anode), v_d (drain/base), and v_e (emitter). The three input signals v_{cd_old} , v_{ed_old} , v_{gs_old} and their three output counterparts v_{cd_new} , v_{ed_new} , v_{gs_new} are used to transfer the necessary history values between successive Newton iterations.

As shown in Fig. 8, the IGBT hardware module executes in four stages. In Stage 1, the two junction limit units and the charge limit unit run in parallel. In Stage 2, eight other units run in parallel, which include the i_{gen} , M , i_{mos} , (i_T, i_{bss}, i_{css}) , C_{gs} , C_{dg} , C_{dsj} unit, and the (C_{cer}, C_{eb}) . In Stage 3, the i_{mult} unit and C_{mult} unit run in parallel. In Stage 4, all the dynamic conductances and equivalent currents calculated by the above units are combined to the output conductance matrix \mathbf{G}^{IGBT} and equivalent current source vector \mathbf{i}_{eq}^{IGBT} . Moreover, the internal structure of all the aforementioned units or subunits are parallelized. The operation sequence of the IGBT hardware module can be seen in Fig. 9, where six different states (S_3 – S_8) contribute to Stage 2. The reason is that W subunit and Q subunit of the (i_T, i_{bss}, i_{css}) unit interconnect to the C_{dsj} unit and the (C_{cer}, C_{eb}) unit, respectively. So for synchronization within the (i_T, i_{bss}, i_{css}) unit, the two subunits responsible for W , p_0 computations are also executed in parallel, these units need to be triggered at the right time.

The internal hardware structure of one of the units, the MOSFET current i_{mos} unit, is shown in Fig. 10, to illustrate the parallelism. There are three comparators and five multiplexers, which are responsible for selecting four groups of results among three nested conditions, see (38) and (39). Since one group of results (when $v_{gs} < V_{th}$) is obtained directly, the remaining three groups of temporary results for i_{mos} , g_{mosgs} , g_{mosds} are all calculated before being selected by the three layers of multiplexers under different conditions as the final

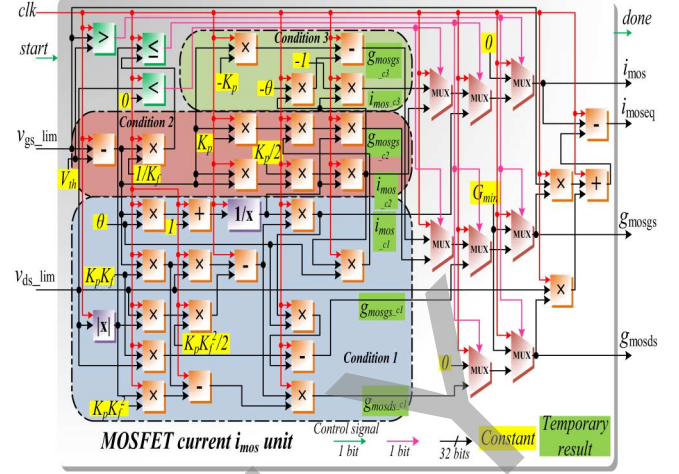


Fig. 10. Hardware structure of the MOSFET current i_{mos} unit.

results. Only after the results for i_{mos} , g_{mosgs} , and g_{mosds} are obtained, the value of i_{moseq} can be computed. In Fig. 10, it is highlighted that the hardware structures to calculate the results under three conditions are fully in parallel. All basic function and nonlinear operator IP cores were generated using the IP Catalog in Xilinx Vivado Design Suite [23].

III. POWER ELECTRONIC CIRCUIT HARDWARE EMULATION

With detailed hardware modules for the power diode, the IGBT and the linear circuit components, the emulation of a complete power converter can be realized. As shown in Fig. 11, it is composed of three intervals within a simulation time-step Δt . Interval 1 is responsible for selecting the proper time-step Δt utilizing the variable time-step control module (VTCM) as well as the input voltage of the converter circuit. Interval 2 is responsible for performing the Newton iterations, which includes the calculations of the conductance matrix and equivalent current source vector for the whole converter circuit and obtaining the solution for circuit node voltages using a parallel linear solver. Although the linear component module is triggered simultaneously with the power diode and IGBT modules, their input node voltages are deliberately kept constant unlike those of the nonlinear ones. The reason to put them inside the Newton iteration loop is to increase the parallelism of the whole structure so as to reduce latency. Most of the Δt history terms and Newton history terms are related to the modules in this interval, which are updated during the calculations between successive time-steps and Newton iterations, respectively.

A. Variable Time-Step Control and Output Modules (VTCMs and VTOM)

Accuracy and speed are both essential for the power converter hardware emulation. The time-step Δt could be large when the circuit is operating under steady state, thereby gaining speed; whereas Δt should be small during transients for higher accuracy. The VTCM in Interval 1 adjusts the time-step

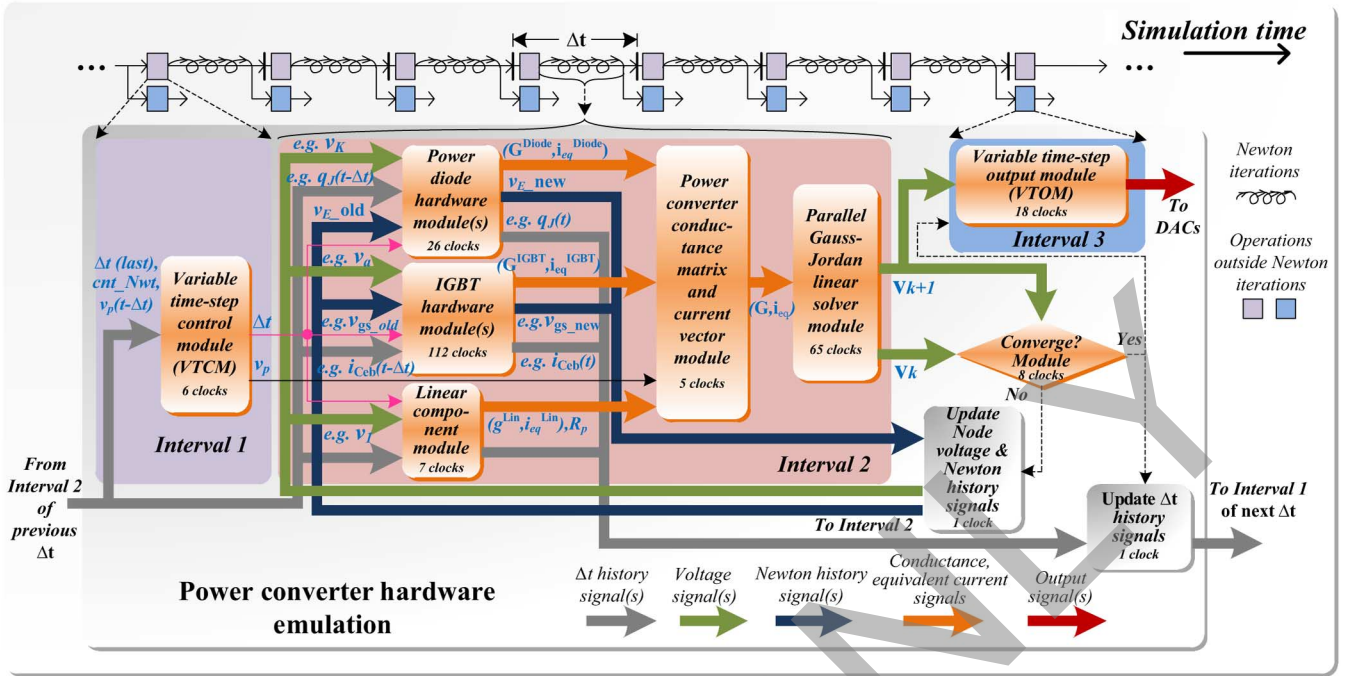


Fig. 11. Sequence of three intervals within a time-step Δt of the power converter hardware emulation.

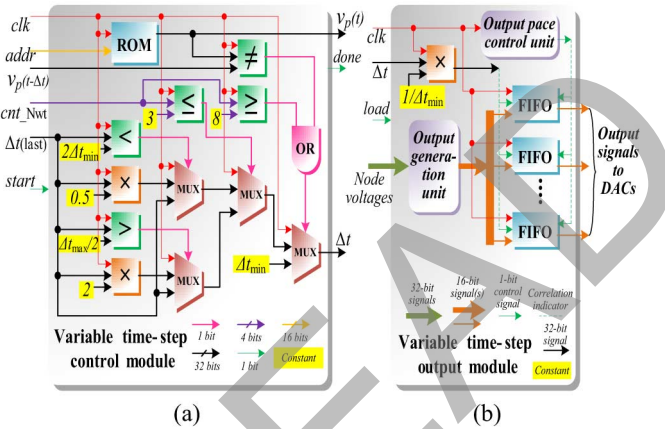


Fig. 12. Hardware structures of (a) VTCM and (b) VTOM modules.

Δt based on the number of Newton iterations cnt_Nwt during the calculations of the last time-step. When $cnt_Nwt \leq 3$, the time-step increased by 2, and when $cnt_Nwt > 3$, the time-step is decreased by 2. The main function of the VTOM is to ensure that the required output signals are calculated from the converged node voltages, and they are output at a fixed sample rate to guarantee that the waveforms are not out of shape. It receives uneven calculation data from the Newton iterations and sends the outputs at equidistant intervals of time. Fig. 12 shows the hardware architectures of the VTCM and VTOM modules.

The input switching signal of the IGBT v_p was precalculated and put into a ROM. At the beginning of an emulation step, it is compared with its history value from the last time-step $v_p(t - \Delta t)$ to decide whether to set the current time-step to minimum. Then, a group of nested comparators and three layers of multiplexers adjust the current time-step for efficiency.

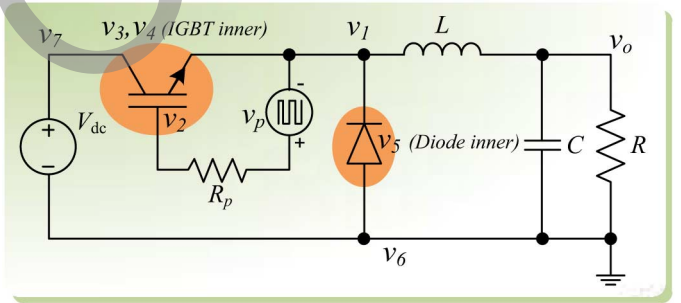


Fig. 13. DC-DC buck converter test circuit.

TABLE I
HARDWARE RESOURCES UTILIZED BY CIRCUIT COMPONENTS

Resource	Diode	IGBT	Overall circuit
FF (flip flop)	7763 (1.3%)	61 384 (10%)	72 181 (12%)
LUT (lookup table)	10 741 (3.5%)	88 061 (29%)	107 264 (35%)
Memory LUT	86 (0.066%)	603 (0.46%)	729 (0.56%)
I/O	32 (4.6%)	32 (4.6%)	91 (13%)
BRAM (block RAM)	1 (0.049%)	34 (1.7%)	38 (1.8%)
DSP48	158 (5.6%)	1419 (51%)	1657 (59%)
BUFG (global buffer)	0 (0.0%)	0 (0.0%)	1 (3.1%)

In the VTOM module, the output pace of the results is based on the current time-step Δt by multiplying a coefficient $\frac{1}{\Delta t_{min}}$ to get an integer value, which in turn controls the inputs of a group of first in first out (FIFO) registers. The larger the current time-step, the more duplicated output results being stored in those FIFOs. The output pace control unit is designed to adjust the refresh rate of the FIFOs, which can be specified by the user. Consequently, the results are sent out evenly with respect to time.

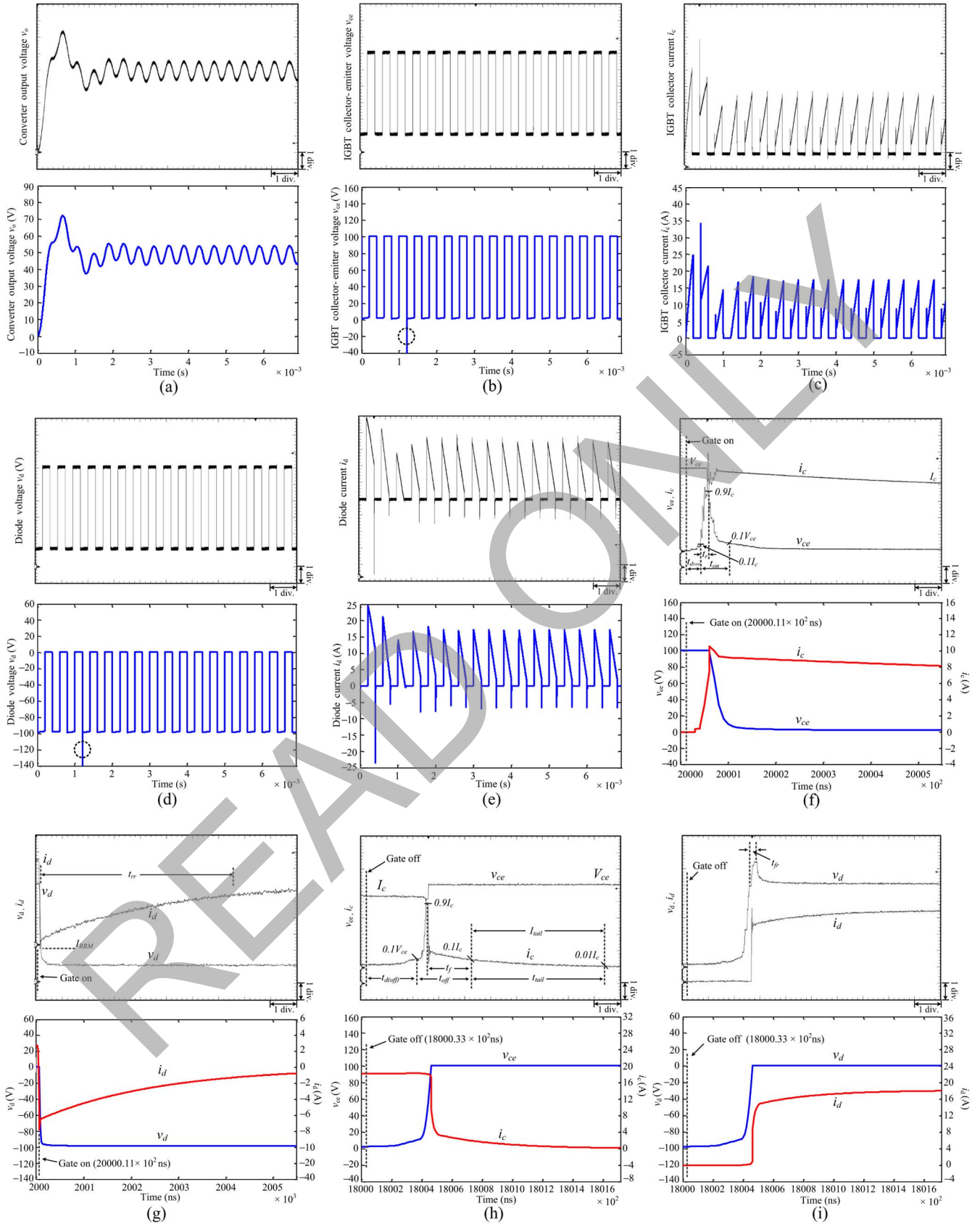


Fig. 14. Steady-state and transient results for dc-dc converter from hardware emulation (oscilloscope) and offline simulation (Saber software) at 2.5 kHz switching frequency. Scale: (a) y -axis: 10 V/div.; (b) y -axis: 20 V/div.; (c) y -axis: 5 A/div.; (d) y -axis: 20 V/div.; (e) y -axis: 5 A/div.; (f) y -axis: 20 V/div. (v_{ce}) and 2 A/div. (i_c); (g) 20 V/div. (v_d) and 2 A/div. (i_d); (h) y -axis: 20 V/div. (v_{ce}) and 4 A/div. (i_c); (i) y -axis: 20 V/div. (v_d) and 4 A/div. (i_d); (a)–(e) x -axis: 4.0 ms; (f) x -axis: 320 ns; (g) x -axis: 3.2 μ s; (h) and (i) x -axis: 1 μ s.

TABLE II
COMPARISON OF DEVICE SWITCHING TIMES AND POWER DISSIPATION
AT 2.5 KHz SWITCHING FREQUENCY

	Saber	FPGA	Error (%)
Switching times			
$t_{d(\text{ON})}(\text{IGBT})$	28.8 ns	29.5 ns	2.43
$t_{\text{ON}}(\text{IGBT})$	60.0 ns	58.7 ns	2.17
$t_r(\text{IGBT})$	18.6 ns	17.8 ns	4.30
$t_{d(\text{OFF})}(\text{IGBT})$	327 ns	334 ns	2.14
$t_{\text{OFF}}(\text{IGBT})$	380 ns	368 ns	3.16
$t_{\text{tail}}(\text{IGBT})$	860 ns	879 ns	2.21
$t_f(\text{IGBT})$	285 ns	293 ns	2.81
$t_{\text{rr}}(\text{Diode})$	4009 ns	4020 ns	0.27
$t_{\text{fr}}(\text{Diode})$	n/a	42.1 ns	n/a
Dissipated Power			
$P_{\text{ON}}(\text{IGBT})$	96.05 mW	93.87 mW	2.27
$P_{\text{OFF}}(\text{IGBT})$	539.6 mW	522.5 mW	3.17
$P_{\text{cond}}(\text{IGBT})$	11.46 W	11.05 W	3.58
$P_{\text{rr}}(\text{Diode})$	5.434 W	5.507 W	1.34
$P_{\text{cond}}(\text{Diode})$	4.768 W	4.782 W	0.29

B. Newton Iterations

Given an n -dimensional nonlinear power converter circuit represented as

$$\dot{\mathbf{i}} = F(\mathbf{v}) \quad (42)$$

where $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$ and $\mathbf{i} = (i_1, i_2, \dots, i_n)^T$ are node voltages and current injection vectors, respectively, and $F(\cdot)$ is the general nonlinear operator. Using classical Newton–Raphson, the node voltage vector at $(k+1)$ th iteration are calculated as

$$\mathbf{G}(\mathbf{v}_k) \cdot \mathbf{v}_{k+1} = \mathbf{i}_{\text{eq}}(\mathbf{v}_k) \quad (43)$$

where $\mathbf{i}_{\text{eq}}(\mathbf{v}_k)$ is the vector of equivalent current sources, and $\mathbf{G}(\mathbf{v}_k)$ is the general linearized conductance matrix given as

$$\mathbf{G}(\mathbf{v}_k) = \begin{pmatrix} \frac{\partial i_1}{\partial v_1} \Big|_{\mathbf{v}_k} & \frac{\partial i_1}{\partial v_2} \Big|_{\mathbf{v}_k} & \dots & \frac{\partial i_1}{\partial v_n} \Big|_{\mathbf{v}_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial i_n}{\partial v_1} \Big|_{\mathbf{v}_k} & \frac{\partial i_n}{\partial v_2} \Big|_{\mathbf{v}_k} & \dots & \frac{\partial i_n}{\partial v_n} \Big|_{\mathbf{v}_k} \end{pmatrix}. \quad (44)$$

The generalized $[\mathbf{G}(\mathbf{v}_k), \mathbf{i}_{\text{eq}}(\mathbf{v}_k)]$ pair for the whole converter circuit can be directly obtained by superimpositions from the those of its separate linear and nonlinear components, such as $(\mathbf{G}^{\text{Diode}}, \mathbf{i}_{\text{eq}}^{\text{Diode}})$ and $(\mathbf{G}^{\text{IGBT}}, \mathbf{i}_{\text{eq}}^{\text{IGBT}})$, which will be illustrated in the case study. The iteration convergence criteria is given as

$$\left| \frac{v_{i(k+1)} - v_{i(k)}}{v_{i(k)}} \right| \leq \epsilon \quad (\epsilon = 10^{-3}) \quad (i = 1, 2, \dots, n). \quad (45)$$

C. Parallel Gauss–Jordan Linear Solver

As seen in Fig. 11, in Interval 2, inside the k th Newton iteration, a set of linear equations need to be solved to obtain the node voltages (43). The parallel Gauss–Jordan linear solver module consists of two main stages: 1) forward elimination; and 2) backward substitution, and its hardware implementation is shown in [32]. To improve its latency, the upgraded structure is featured by deep parallelism in both elimination and backward substitution at the cost of using extra hardware resources,

and the pivoting is sped-up to as fast as two clock cycles each step. The advantages of this approach is reduced latency and high efficiency for handling larger matrices.

IV. CASE STUDY AND VALIDATION

A. Test Circuit

The test circuit used for the power converter hardware emulation is a dc–dc buck converter (Fig. 13). There are two nonlinear components (IGBT and Diode), four linear components, and a voltage source in the circuit. The parameters of the circuit are given in the Appendix. The duty ratio of the square wave PWM v_p to trigger the IGBT is 0.5. The IGBT node voltages v_c, v_g, v_a, v_d , and v_e are also the circuit node voltages v_1, v_2, v_7, v_3 , and v_4 , respectively. The diode node voltages v_A, v_{in} , and v_K correspond to the circuit node voltages v_6, v_5 , and v_1 , respectively. The linear components, R, L , and C , can be collectively discretized as a conductance g^{Lin} and a parallel equivalent current source $i_{\text{eq}}^{\text{Lin}}$. Similarly, v_p with a serial resistor R_p can be equally changed to a current source $\frac{v_p}{R_p}$ and a parallel resistor R_p in order to reduce the dimension of the whole system [33]. The dimension of the discrete-time linearized system is originally 7, but since the circuit node voltages $v_6 = 0$ (ground) and $v_7 = V_{\text{dc}}$, it was reduced to 5. Thus the discrete-time linearized equation for the converter circuit can be expressed as (49) in the Appendix.

This dc–dc converter was emulated on the Xilinx Virtex-7 XC7VX485 T FPGA board, which was connected to a 16-bit four channel digital to analog converters (DACs). This board has 607 200 FFs, 303 600 LUTs, 130 800 Memory LUTs, 700 I/Os, 2060 BRAMs, 2800 DSP48 s, and 32 BUFGs. The emulation results were captured by a four-channel oscilloscope. The offline simulation for the dc–dc converter was executed on a PC with Intel Core™ i7-2600 K 3.4 GHz CPU, 8 GB RAM, running Windows 7 operating system. The execution time for the offline simulation of the converter for 100 ms of run time in Saber using variable time-step strategy (with an initial time-step of 10 ns and maximum time-step of 1 μ s) was 17.5 s, while the hardware emulation time was 0.58 s. Therefore, the speedup is more than 30 times. It is conceivable that offline simulation of converter circuits with more IGBTs and diodes employing detailed device-level models would be much slower than hardware emulation. The latency of one time-step calculation is 579 clock cycles, the highest frequency is 115 MHz, and the resource utilization is listed in Table I.

B. Results and Comparison

1) *Time-Domain Results*: The steady-state results for the converter output voltage v_o , the IGBT collector–emitter voltage $v_{ce}(=v_7 - v_1)$, the IGBT collector current i_c , the diode voltage $v_d(=v_6 - v_1)$, and the diode current i_d of the dc–dc converter hardware emulation under the switching frequency of 2.5 kHz are shown in Fig. 14(a)–(e). Compared with Saber, the steady-state results from hardware emulation are accurate. Since Saber did not converge when R_{M0} is set to 50 Ω , the waveforms shown in Fig. 14(a)–(h) were obtained by

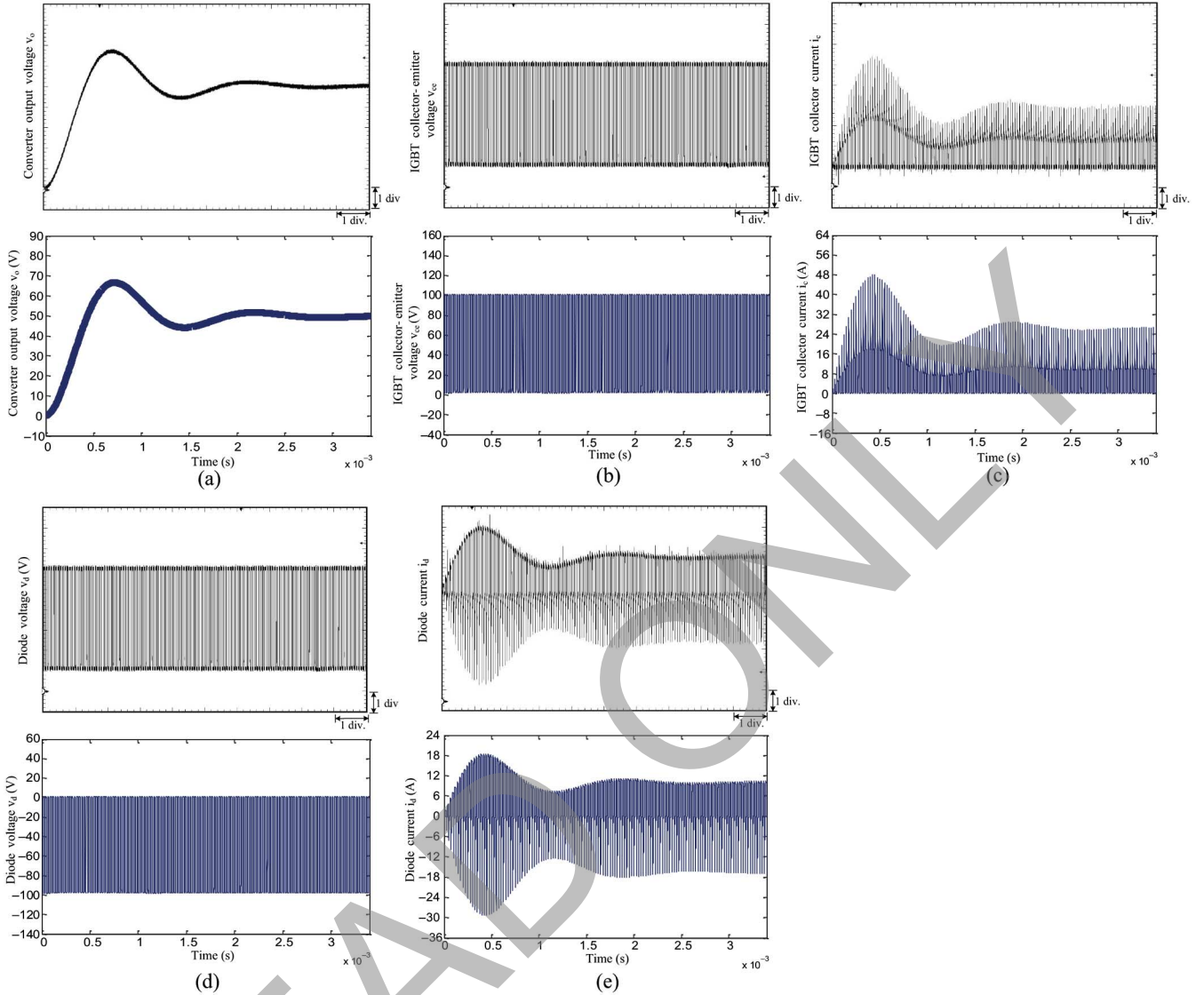


Fig. 15. High-frequency switching results at 40 kHz for the dc-dc converter from hardware emulation (oscilloscope) and offline simulation (Saber software). Scale: (a) y -axis: 10 V/div.; (b) y -axis: 20 V/div.; (c) y -axis: 8 A/div.; (d) y -axis: 20 V/div.; (e) y -axis: 6 A/div.; (a)–(e) x -axis: 12.5 ms.

setting R_{M0} to 0Ω (no forward recovery). The dash-circled areas in Fig. 14(b) and (d) are the nonconvergence points of Saber.

The device-level transient results for (v_{ce}, i_c) and (v_d, i_d) during turn-ON and turn-OFF switching of the dc-dc converter hardware emulation under the switching frequency of 2.5 kHz are shown in Fig. 14(f)–(i). Compared with Saber, the device-level transient results of the hardware emulation show good agreement. Specifically, the transient waveform of diode voltage from oscilloscope in Fig. 14(i) was obtained by setting R_{M0} to 50Ω , which shows the forward recovery clearly. The comparison of IGBT and diode switching times from Saber and hardware emulation is shown in Table II. The hardware emulation was also tested under high-switching frequency conditions, up to a maximum of 100 kHz. The results for a 40 kHz switching frequency are shown in Fig. 15. Again, a close agreement with Saber can be observed.

2) *Power Dissipation Analysis:* The power dissipation during a switching cycle of IGBT mainly comes from the switching

loss and conduction loss. The instantaneous power dissipation is calculated by multiplying the IGBT collector-emitter voltage (v_{ce}) and collector current (i_c). By choosing corresponding time range from t_0 to t_1 , the energy losses in one switching cycle are obtained by integrating the instantaneous power dissipation during turn-ON, turn-OFF, and conduction period, respectively. The power loss P_{loss} is quotient of the energy loss and switching period T_s , expressed as

$$P_{\text{loss}} = \frac{\int_{t_0}^{t_1} v_{ce}(t) \cdot i_c(t) dt}{T_s}. \quad (46)$$

The power dissipation of reverse recovery and conduction period for the diode are obtained similarly. Table II shows the power dissipation results under the switching frequency of 2.5 kHz from both Saber and hardware emulation. The relative errors between the hardware emulation and Saber have a few reasons. The first can be attributed to the modeling error. Specifically, the power diode model used by Saber is

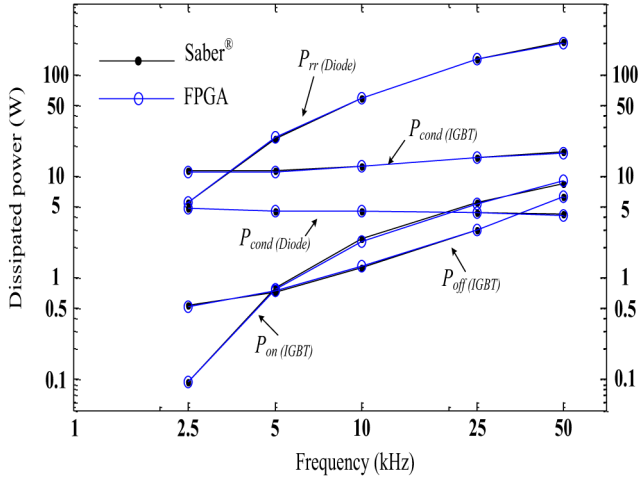


Fig. 16. Variation of device power dissipation with switching frequency from Saber and hardware emulation.

more sophisticated than the one in this work. For the IGBT model, although both Saber and the hardware emulation use Hefner's model, Saber's model is more refined judging from its larger parameter set. Another reason for the error could be the limitation of the 32-bit floating-point number precision in the numerical solvers used in the hardware emulation of the highly complicated physics-based device-level models,

which may have resulted in a loss of accuracy. Nevertheless, the variation of device power dissipation with switching frequency (Fig. 16) shows close agreement between the emulation and Saber both at low as well as high-switching frequencies proving the high-fidelity of the detailed hardware emulation. As the switching frequency increased, as expected, the IGBT switching losses ($P_{on(IGBT)}$, $P_{off(IGBT)}$) and diode reverse recovery loss ($P_{rr(Diode)}$) increased significantly; however, the IGBT conduction loss ($P_{cond(IGBT)}$) increased much slowly, and the diode conduction loss ($P_{cond(Diode)}$) remained almost constant.

V. CONCLUSION

This work proposed a digital hardware emulation of detailed physics-based device-level IGBT and diode models. Both these device-level models are fully paralleled on the FPGA. The hardware emulation of both devices is based on a unified numerical framework based on Newton linearization, and can be extended in a straightforward fashion to model complete power electronics circuits. The variable time-step strategy makes the hardware emulations with power converter based on physics-based analytical model become possible, which not only ensures the speed of the emulation but also the detailed switching behaviors. Therefore, the emulated models were able to predict the dissipated energy in the power electronic circuit quite accurately. Comparison with Saber proves that the emulated

$$\mathbf{G}^{IGBT} = \begin{bmatrix} g_{Cgs} + g_{Cdsj} + g_{Ccerbc} + g_{mosds} + g_{mosgs} + g_{multgs} + g_{multds} + g_{Cmultbc} + g_{cssbc} & -g_{Cgs} - g_{mosgs} & -g_{cssae} & -g_{Cdsj} - g_{mosds} - g_{Ccerbc} & g_{cssae} - g_{csseb} + g_{multae} - g_{multeb} \\ g_{mosds} + g_{mosgs} + g_{multgs} + g_{multds} + g_{Cmultbc} + g_{cssbc} & -g_{multgs} & -g_{multae} + g_{csseb} - g_{multds} + g_{multeb} & -g_{Cmultbc} - g_{cssbc} & g_{multae} - g_{multeb} \\ -g_{Cgs} & g_{Cgs} + g_{Cdg} & 0 & -g_{Cdg} & 0 \\ -g_{Tbc} & 0 & g_{Tae} & -g_{Teb} + g_{Tbc} & g_{Teb} - g_{Tae} \\ g_{bssbc} - g_{Cdsj} - g_{mosds} & -g_{Cdg} + g_{mosgs} & g_{Cdg} + g_{Cdsj} + g_{Ceb} & -g_{Ceb} - g_{bsseb} & -g_{multae} + g_{multeb} \\ -g_{mosgs} - g_{multgs} & +g_{multgs} & g_{multae} & +g_{mosds} + g_{bsseb} & -g_{multae} + g_{multeb} \\ -g_{multds} - g_{Cmultbc} & & +g_{multds} - g_{multeb} & & \\ -g_{Ccerbc} - g_{bssbc} & & g_{cssae} - g_{multae} & -g_{Ceb} - g_{bsseb} + g_{Ccerbc} & g_{Ceb} + g_{bsseb} \\ -g_{cssbc} + g_{Tbc} & 0 & g_{multae} & -g_{csseb} + g_{Teb} + g_{bssbc} + g_{cssbc} - g_{Tbc} & -g_{cssae} + g_{csseb} - g_{Teb} + g_{Tae} \end{bmatrix} \quad (47)$$

$$\mathbf{i}_{eq}^{IGBT} = \begin{bmatrix} i_{moseq} + i_{csseq} + i_{multeq} & -i_{moseq} + i_{bsseq} - i_{multeq} & -i_{bsseq} - i_{csseq} \\ +i_{Cgseq} + i_{Cdsjeq} & -i_{Cgseq} + i_{Cdgeq} & -i_{Teq} & i_{Cdsjeq} - i_{Cdgeq} & -i_{Cebseq} - i_{Ccereq} \\ +i_{Ccereq} + i_{Cmulteq} & & +i_{Cebseq} - i_{Cmulteq} & & +i_{Teq} \end{bmatrix}^T \quad (48)$$

$$\begin{bmatrix} \mathbf{G}^{IGBT}(1,1) + \mathbf{G}^{IGBT}(1,2) & \mathbf{G}^{IGBT}(1,4) & \mathbf{G}^{IGBT}(1,5) & \mathbf{G}^{Diode}(3,2) \\ \mathbf{G}^{Diode}(3,3) + -1/R_g & & & \\ g^{Lin} + 1/R_g & & & \\ \mathbf{G}^{IGBT}(2,1) & \mathbf{G}^{IGBT}(2,2) & \mathbf{G}^{IGBT}(2,4) & \mathbf{G}^{IGBT}(2,5) & 0 \\ -1/R_g & +1/R_g & & & \\ \mathbf{G}^{IGBT}(4,1) & \mathbf{G}^{IGBT}(4,2) & \mathbf{G}^{IGBT}(4,4) & \mathbf{G}^{IGBT}(4,5) & 0 \\ \mathbf{G}^{IGBT}(5,1) & \mathbf{G}^{IGBT}(5,2) & \mathbf{G}^{IGBT}(5,4) & \mathbf{G}^{IGBT}(5,5) & 0 \\ \mathbf{G}^{Diode}(2,3) & 0 & 0 & 0 & \mathbf{G}^{Diode}(2,2) \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \begin{bmatrix} i_{eq}^{IGBT}(1) + i_{eq}^{Diode}(3) - i_{eq}^{Lin} \\ -v_p/R_g - \mathbf{G}^{IGBT}(1,3) \cdot V_{dc} \\ i_{eq}^{IGBT}(2) + v_p/R_g - \mathbf{G}^{IGBT}(2,3) \cdot V_{dc} \\ i_{eq}^{IGBT}(4) - \mathbf{G}^{IGBT}(4,3) \cdot V_{dc} \\ i_{eq}^{IGBT}(5) - \mathbf{G}^{IGBT}(5,3) \cdot V_{dc} \\ i_{eq}^{Diode}(2) \end{bmatrix} \quad (49)$$

models are both accurate and efficient in terms of speed of execution. While the hardware emulation is still slower than real-time execution to be applied for closed-loop HIL simulation, with newer FPGA generations, and further efficiency refinements in the numerical solution that goal appears feasible. Presently, the proposed hardware models can be readily used for open-loop HIL applications and for circuit design acceleration that involve statistical, parametric, and sensitivity analyses.

APPENDIX

Test circuit parameters: $V_{dc} = 100\text{ V}$, $R = 5\ \Omega$, $L = 700\ \mu\text{H}$, $C = 70\ \mu\text{F}$, $R_g = 100\ \Omega$.

IGBT parameters: Out of 32 parameters, the ones used in this paper are: $K_p = 1\text{ A/V}$, $K_f = 2$, $\theta = 0.01\text{ V}^{-1}$, $V_{th} = 5\text{ V}$, $N_B = 2 \times 10^{14}\text{ cm}^{-3}$, $I_{sne} = 10^{-14}\text{ A}$, $n_i = 1.45 \times 10^{10}\text{ cm}^{-3}$, $\tau_{HL} = 4 \times 10^{-7}\text{ s}$, $A = 0.1\text{ cm}^2$, $q = 1.6 \times 10^{-19}\text{ C}$, $L = 2.7 \times 10^{-3}\text{ cm}$, $T = 296\text{ K}$, $k = 8.617 \times 10^{-5}\text{ eV/K}$, $\beta = 0.4615$, $\epsilon_{si} = 1.05 \times 10^{-12}\text{ F/cm}$, $W_B = 0.01\text{ cm}$, $b = 3.33$, $D_p = 11.48\text{ cm}^2/\text{s}$, $BV_n = 4$, $BV_{cb0} = 3.18 \times 10^7\text{ V}$, $G_{min} = 10^{-12}\text{ S}$, $A_{gd} = 0.05\text{ cm}^2$. The rest can be obtained from Saber.

Diode parameters: $I_S = 10^{-14}\text{ A}$, $\tau = 5\ \mu\text{s}$, $T_M = 5\ \mu\text{s}$, $V_T = 0.0259\text{ V}$, $m = 0.5$, $C_{J0} = 1\text{ nF}$, $V_J = 0.7\text{ V}$, $I_{SE} = 10^{-22}\text{ A}$, $R_{M0} = 50\ \Omega$, $R_C = 10^{-3}\ \Omega$.

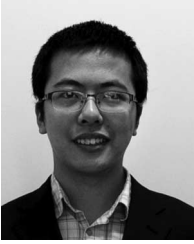
REFERENCES

- [1] H. Li, M. Steurer, S. Woodruff, K. L. Shi, and D. Zhang, "Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real time simulation," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1144–1151, Jun. 2006.
- [2] A. Bouscayrol, X. Guillaud, R. Teodorescu, P. Delarue, and B. Lemaire-Semail, "Energetic macroscopic representation and inversion-based control illustrated on a wind-energy-conversion system using hardware-in-the-loop simulation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4826–4835, Dec. 2009.
- [3] M. O. Faruque and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1146–1158, Apr. 2010.
- [4] D. Majstorovic, I. Celanovic, N. D. Teslic, N. Celanovic, and V. A. Katic, "Ultralow-latency hardware-in-the-loop platform for rapid validation of power electronics designs," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4708–4816, Oct. 2011.
- [5] E. Adzic, M. Adzic, V. Katic, D. Marcetic, and N. Celanovic, "Development of high-reliability EV and HEV IM propulsion drive with ultra-low latency HIL environment," *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 630–639, May 2013.
- [6] Y. Chen and V. Dinavahi, "Hardware emulation building blocks for real-time simulation of large-scale power grids," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 373–381, Feb. 2014.
- [7] G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.*, vol. 22, no. 2, Apr. 2007, pp. 1235–1246.
- [8] A. Myaing and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics," *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 358–368, Jan. 2011.
- [9] E. Monmasson *et al.*, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.
- [10] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1300–1309, Feb. 2012.
- [11] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems," *IET Gen. Transmiss. Distrib.*, vol. 7, no. 5, pp. 451–463, May 2013.
- [12] Z. Ma, J. Gao, and R. Kennel, "FPGA implementation of a hybrid sensorless control of SMPMSM in the whole speed range," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1253–1261, Aug. 2013.
- [13] M. Brox *et al.*, "CAD tools for hardware implementation of embedded fuzzy systems on FPGAs," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1635–1644, Aug. 2013.
- [14] O. Jimenez, O. Lucia, L. A. Barragan, J. I. Artigas, and I. Urriza, "FPGA-based test-bench for resonant inverter load characterization," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1645–1654, Aug. 2013.
- [15] I. Bahri, L. Idkhajine, E. Monmasson, and M. El Amine Benkhelifa, "Hardware/software codesign guidelines for system on chip FPGA-based sensorless AC drive applications," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2165–2176, Nov. 2013.
- [16] Y. Wang and V. Dinavahi, "Low-latency distance protective relay on FPGA," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 896–905, Mar. 2014.
- [17] 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, Jul. 2014.
- [18] A. R. Hefner and D. M. Diebolt, "An experimentally verified IGBT model implemented in the Saber circuit simulator," *IEEE Trans. Power Electron.*, vol. 9, no. 5, pp. 532–542, Sep. 1994.
- [19] A. R. Hefner, "Modeling buffer layer IGBTs for circuit simulation," *IEEE Trans. Power Elect.*, vol. 10, no. 2, pp. 111–123, Mar. 1995.
- [20] R. Kraus and K. Hoffmann, "An analytical model of IGBT's with low emitter efficiency," in *Proc. 5th Int. Symp. Power Semicond. Devices. (ISPSD)*, Monterey, CA, USA, 1993, vol. 3, pp. 30–34.
- [21] P. O. Lauritzen and C. L. Ma, "A simple diode model with reverse recovery," *IEEE Trans. Power Electron.*, vol. 6, pp. 188–191, Apr. 1991.
- [22] C. L. Ma and P. O. Lauritzen, "A simple power diode model with forward and reverse recovery," *IEEE Trans. Power Electron.*, vol. 8, no. 4, pp. 342–346, 1993.
- [23] Xilinx Inc. *Ultra Fast Design Methodology Guide for the Vivado Design Suite*. Xilinx, Inc., Apr. 2014.
- [24] J. T. Hsu and K. D. Ngo, "Behavioral modeling of the IGBT using the Hammerstein configuration," *IEEE Trans. Power Electron.*, vol. 11, no. 6, pp. 746–754, Nov. 1996.
- [25] C. Wong, "EMTP modeling of IGBT dynamic performance for power dissipation estimation," *IEEE Trans. Ind. Appl.*, vol. 33, no. 1, pp. 64–71, Jan./Feb. 1997.
- [26] B. Allard, H. Morel, S. Ghedira, and A. Ammous, "Building advanced averaged models of power converters from switched bond graph representation," *Soc. Comput. Simul., Sim. Ser.*, vol. 31, no. 1, pp. 331–338, 1999.
- [27] B. J. Baliga, "Analytical modeling of IGBTs: Challenges and solutions," *IEEE Trans. Electron. Devices*, vol. 60, no. 2, pp. 535–543, Feb. 2013.
- [28] K. Sheng, B. W. Williams, and S. J. Finney, "A review of IGBT models," *IEEE Trans. Power Electron.*, vol. 15, pp. 1250–1266, Nov. 2004.
- [29] H. Garrab *et al.*, "On the extraction of PiN diode design parameters for validation of integrated power converter design," *IEEE Trans. Power Electron.*, vol. 20, no. 3, pp. 660–670, May 2005.
- [30] F. Mohammed *et al.*, "A novel silicon schottky diode for NLTL applications," *IEEE Trans. Electron Devices*, vol. 52, no. 7, pp. 1384–1391, Jul. 2005.
- [31] G. T. Oziemkiewicz, "Implementation and development of the NIST IGBT model in a SPICE-based commercial circuit simulator," M.Sc. thesis, Dept. Elect. Comput. Eng., Univ. Florida, Gainesville, FL, USA, 1995.
- [32] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. Ind. Electron.*, pp. 1–8, 2010.
- [33] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1975.



Wentao Wang (S'14) received the B.Eng. and M.Sc. degrees in automation engineering from the Qingdao University of Science and Technology, Qingdao, China, in 2002 and 2005, respectively. Currently, he is pursuing the M.Sc. degree in electrical and computer engineering at the University of Alberta, Edmonton, AB, Canada.

He worked as a Software Engineer with Shanghai Automatic Instrumentation Co. Ltd., (SAIC), China. His research interests include real-time simulation of power systems, power electronics device modeling, and field-programmable gate array design.



Zhuoxuan Shen (S'14) received the B.Eng. degree in electrical engineering from Jiangsu University, Zhenjiang, China, in 2013. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of Alberta, Edmonton, AB, Canada.

His research interests include real-time simulation of power systems, power electronics, and field-programmable gate arrays.



Venkata Dinavahi (S'94–M'00–SM'08) received the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000.

He is currently a Professor in Electrical and Computer Engineering with the University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronic systems, large-scale system simulation, and parallel and distributed computing.

READ ONLY