FPGA, PHYSICS-BASED MODELING OF IGBT AND PIN DIODE FOR HARDWARE CO-SIMULATION OF COMPLEX POWER ELECTRONIC CONVERTERS AND SYSTEMS

P. Aristidou^(a), P.R. Palmer^(b)

^(a)Department of Engineering University of Cambridge, Cambridge, UK ^(b) Department of Engineering University of Cambridge, Cambridge, UK

(a)pa297@cam.ac.uk, (b)prp@eng.cam.ac.uk

ABSTRACT

This paper presents the steps and the challenges for implementing analytical, physics-based models for the insulated gate bipolar transistor (IGBT) and the PIN diode in hardware and more specifically in field programmable gate arrays (FPGAs). The models can be utilised in hardware co-simulation of complex power electronic converters and entire power systems in order to reduce the simulation time without compromising the accuracy of results. Such a co-simulation allows reliable prediction of the system's performance as well as accurate investigation of the power devices' behaviour during operation. Ultimately, this will allow application-specific optimisation of the devices' structure, circuit topologies as well as enhancement of the control and/or protection schemes.

Keywords: FPGA, hardware co-simulation, IGBT, power device analytical electro-thermal modeling

1. INTRODUCTION

In the field of power electronics, circuit and system computer simulation has become a vital tool for developing successful designs, first prototypes and final products with minimal experimental work. Ideally such a simulation needs to be sufficiently accurate in order to permit a reliable examination of the detailed circuit/system operation and fast enough in order to allow quick identification of the required performance trade-offs, and thereby be utilised for design optimisation. Satisfying both of the above requirements for any level of design complexity, however, is still a major challenge that today with the increased application of power electronic converters in large scale power systems, exercises the power electronics industry more than ever before.

There are many reasons accounting for this great challenge. First of all the characteristic times within a power electronic system may be different by many orders of magnitude, ranging from nanoseconds and microseconds for the switching transitions of the power semiconductor devices and the typical switching cycles, through several seconds for the load or fault transient responses to several minutes/hours/days for complete load/thermal cycles (Pejovic and Maksimovic 1994). Thus simulating the full time-domain response of a power electronic circuit/system with high resolution may require trillions of time steps to be calculated. In addition to this, the simulation should be able to faithfully capture the switching behaviour of the power devices via the use of accurate/detailed models and robust enough for handling their inherent nonlinearities. At present, implementing such a long-time, detailed simulation is infeasible or -in the best case- extremely time consuming and therefore impractical.

Generally the required time for each time step computation scales with the detail of the power devices representation as well as the circuit/system complexity; hence depending on the objective of the simulation, different devices' models, software packages and circuits are being used in order to make it practical. On the one extreme we have the numerical models constructed in 2D or 3D finite differences or finite element packages, such as Silvaco Atlas and Sentaurus Device. These models can provide a precise picture of the devices' physical phenomena and can be utilised in mixed-mode circuit simulations to give accurate results regarding the devices' switching operation, namely power losses, transition durations and voltage/current overshoots. This kind of simulation however, is extremelv computationally expensive typically requiring several minutes or even hours to simulate just few switching cycles (~100-200µs) of simple test circuits comprised only of one or two power devices. As such this approach is of limited applicability for circuit/system designers and is mostly restricted to device manufacturers whose objective is mainly the enhancement of general device characteristics. However, even for this latter application there is difficulty in drawing useful conclusions about the effects that several structural parameters have on the device's performance since the link of the parameters with the simulation output is purely numerical. On the other extreme we have the behavioural and average device or converter models (e.g. Oh and Nokali 2001, Jin 1997) that simulate a device or converter behaviour based on databases, curves, expressions or components fitted to experimental or datasheet output device or converter characteristics. These models can be constructed in any circuit or equation solver package

and their use can significantly speed up the simulation, allowing thereby investigation of complex circuits/systems or long load/thermal cycles. However the results lack any sort of accuracy and are thus of limited creditability. Even when a behavioural model claims giving precise results for the certain circuit topology used for its experimental extraction, its validity in different or more complex topologies is extremely limited due to the change of the values of several stray components (e.g. capacitances and inductances). Furthermore, since these models do not consider any device physics or internal parameters, they cannot be used for device optimisation.

Clearly the quest for general-application accurate and computationally efficient power semiconductor device models has a solution lying somewhere in the middle. More specifically the models should be physics based (as opposed to behavioural models) and analytical (as opposed to numerical models). However, especially for the case of high voltage power bipolar devices, such as the PIN diode and the IGBT, there has been a great difficulty in constructing truthful analytical models. This is mainly due to the distributed nature of the charge transport which combined with the large base thickness (required to achieve high voltage capability) amplifies the effects of charge dynamics and thereby makes any simple charge control approaches failing (Leturcq 1997). On the other hand, however, solving analytically the full set of semiconductor charge transport equations is impossible. In order to evade the issue the full equations' set is simplified into one equation: the ambipolar diffusion equation (ADE); which is solvable and still retains the essentials of the distributed nature of charge transport. The only way to solve this equation without assuming an initial solution shape or making oversimplifications is a Fourier-based one, proposed by Leturcq (Leturcq 1997). With this approach it became possible to develop and validate accurate and fairly robust electrothermal, circuit simulator (e.g. PSpice) and Simulink p-i-n diode and IGBT models [Palmer et al. 2003, Bryant et al. 2007a], together with easy parameter extraction procedures [Bryant et al. 2007b]. The use of these compact models enables reasonably fast simulation times: for the inductive chopper circuit case, it typically takes 5-10s for every switching cycle (~50µs). This is a great improvement against numerical models (~400 times faster); however for simulations of more than few tens of milliseconds or of more complex circuit topologies, the computational time is still prohibitive.

Nowadays, FPGAs have millions of hardware resources and are capable of performing thousands of operations in parallel, allowing complicated functions to be executed within a single-clock cycle. As such FPGAs can be deployed in computationally intensive simulations or other processes in order to carry out the most intense tasks and thereby increase the overall speed of execution.

This paper presents a new step towards the ultimate solution of the power bipolar device modelling

and simulation challenge, which comes in the form of the FPGA implementation of the abovementioned Fourier-based, analytical models. The second section of the paper provides an overview of the main features of the available FPGA technologies and identifies the benefits, the constraints and the challenges of using FPGAs in our application. The third section provides an explanation of the chosen PIN diode and IGBT physicsbased models along with a detailed solution of the electrothermal ADE. In the fourth section the strategy for implementing the device modeling in parallel FPGA programming is presented while conclusions and plans of future work are being given at the end.

2. FEATURES, BENEFITS, LIMITS AND CHALLENGES OF FPGAs

2.1. FPGA Basics

The basic layout of an FPGA chip is shown in fig.1.



Figure 1- Basic layout of an FPGA

Simplistically this layout can be seen as consisting of three main components. First and most important one is a large two-dimensional array of configurable logic blocks, CLBs, each of which can be programmed to perform a combinatorial and/or sequential operation. The other basic component is a number of input/output blocks (IOBs) at the chip periphery whose purpose is to handle the interfacing between the internal chip resources and the external circuitry, including the signals to and from the CLBs and the signals required for programming the necessary logic configurations. Finally there is a large interconnection network consisting of wires and programmable interconnection matrices (PIs) which are responsible for the connection of the CLBs to the IOBs and/or with each other. (Xilinx Inc. 2012a-b, Altera Corp. 2012a-b)

Besides the above basic features today's FPGAs also include a number of higher level functionality elements such as block RAM blocks, hardwired digital signal processing (DSP) blocks, communication blocks and clock manager systems. These dedicated resources generally speed up the execution of some common functions and save the usage of primitive resources for other application-specific operations.

2.2. Benefits of FPGAs

It is evident from the FPGA architecture that the CLBs (and the embedded DSPs) can be set up to perform arithmetic and logic functions like a microprocessor's arithmetic logic unit (ALU). However, in contrast with microprocessors where the arithmetic logic unit is fixed and configured for general purpose use, the CLBs can be customized to implement only application-specific tasks, resulting in improved computational efficiency and optimum hardware resources use.

Moreover due to the numerous available resources it is possible to construct a vast number of tailored ALUs and use them in parallel in order to enable higher computational throughput and vastlv superior performance. FPGAs can even outperform cluster or massively parallel supercomputers with thousands of CPUs (and thus ALUs) since in FPGAs there are no processor cache misses, there is ultralow latency and the execution of operators is efficiently pipelined via pointto-point interconnects (Zack et al. 2004). As a practical proof of the computational accelerations offered by FPGAs is the experience with several complex algorithms. Comparisons of execution times of FPGA hardware co-simulations against offline simulations on single or multiple processors show that the use of the former results to speed-ups ranging from 10x to 1000x, with the mode value lying within 120-200x (Xilinx Inc. 2009, Gonzalez and Nunez 2009). It should therefore be expected that the use of the FPGA technique in our device modeling application will result to improved computational times as well.

Besides giving higher speeds, FPGAs also consume much less power since they can operate at a lower clock frequency than microprocessors (e.g. at 500MHz instead of 3GHz) and still achieve improved performance due to parallelism. This is very advantageous in our case since the FPGA device models are also intended to be utilized as the basis for real time circuit control, and less power consumption means amongst others lower running costs and improved reliability. Furthermore FPGAs can be considered as cost effective, since their average DSP function cost of less than \$2 is much less than that of DSP devices and no more than that of processors (Zack et al. 2004).

2.3. Limitations and Challenges

From the above it is clear that FPGAs can offer significant benefits, however they also exhibit limitations and challenges which must be considered in order to minimize their effect as much as possible.

2.3.1. Representation of Real Numbers

The first limitation is the use of fixed-point arithmetic instead of floating point one since the latter -though possible to implement- results to highly inefficient usage of the hardware resources. More specifically, in order to enable the representation and handling of the standard IEEE-754 floating point structure of the form $(-1)^{(sign)} x$ (normalized mantissa) x (base) ^(exponent), shown in figure 2 (IEEE 2008), each instruction data input and

output requires the use of special mantissa normalisation / denormalisation circuitry employing many interconnections, multiplexers, shifters and counters, ultimately leading to excessive logic usage and slow clock rates.

sign	exponent	trailing mantissa field	
1 bit	w bits (8 or 11)	t bits (23 or 52)	
M	SB LSB	MSB	LSB

Figure 2 - IEEE-754 floating point number

On the contrary the use of fixed-point arithmetic, where a number is represented as a scaled integer by a fixed number of digits before and after a radix point, is suited for FPGA implementation well since conventional 2's complement can be used with only the location of the radix point required to be specified. However, achieving adequate precision might be challenging. First the quantization noise can be much larger than in single or double precision floating-point arithmetic due to the fact that the represented numbers are now uniformly spaced and possibly implemented with fewer bits. Also fixed point operations can produce results having more bits than the operands resulting to information loss due to rounding, truncation or saturation. Therefore in order to enable high precision with fixed-point arithmetic while at the same time achieving optimal utilization of the hardware resources, the bit width of each variable needs to be optimized for the dynamic range and accuracy of the specific algorithm.

2.3.2. Numerical Integration Considerations

Another challenge in using FPGAs in a simulation of a system like ours is the choice of the most appropriate numerical integration method, classically distinguished by three main properties: (i) fixed time-step size/variable time-step size (ii) explicit/implicit and (iii) single-step/multi-step. The choice should be made after considering accuracy, convergence, stability, and speed issues in conjunction with the co-simulation objective and constraints. Below we present the major considerations. Details about the choices and the overall integration methodology are given later in section 4.

Fixed time-step size methods use a constant step size throughout the simulation, whereas variable ones adjust the step size in every time step depending on the model dynamics, namely reducing the size to increase accuracy when the model's states are changing rapidly and increasing it when they are changing slowly in order to avoid taking unnecessary steps (Greenberg 1998). In variable step size methods the solution's truncation error is estimated in every step computation and is used in order to evaluate the largest allowable step size that will keep the error bounded below some specified limits. This means that in a variable step size technique an additional computational overhead is present in every time step, but this is more than compensated by the reduction in number of steps. For this reason variable step size are the methods of choice, however only for offline co-simulations. For real-time co-simulations, the integration method should use fixedstep size in order to assure synchronisation of the data exchange between the co-simulation nodes and meet the real-time related constraints (i.e the computations to be completed within an interval less than or equal to the simulation clock period). The use of fixed time-step size, however, in an event-based system like ours where discrete events (such as control switching pulses) take place in an otherwise continuous model, introduces inaccuracies whenever these events occur at non-integer multiples of the fixed time step and as the simulation proceeds the accumulated error might become large non-characteristic enough to cause harmonics. switching mistakes and other abnormal behaviour (Strunz 2004). Using a sufficiently small time step size solves the abovementioned problem, but this also requires the computational time of the single step to be less than the step interval. Here the difference of our co-simulation fidelity objective to that of real-time power system platforms should be highlighted. The latter concern only with the simulation of the behaviour and response of the overall power system (e.g. detection of voltage sags or fault propagation) thus their models of the power electronic converters are behavioural with no concern of what is happening at the device level. On the contrary our purpose is to increase the simulation fidelity to the level of the exact device operation. This means that the use of small time step sizes (<100ns) is not only for improved accuracy but a requirement in order to capture the switching waveforms.

Regarding the choice between explicit and implicit methods the decision is taken primarily by stability and stiffness considerations. Explicit methods evaluate a state at the next time step as an explicit function of the state values only up to the current time step, whereas implicit ones evaluate the next time-step state as an implicit function of the state values up to the value of the next time step as well (Greenberg 1998). Therefore for dynamic systems, implicit methods require the solution of a system of equations, meaning a higher computational cost in every time step. On the other hand implicit methods offer much greater stability, producing bounded numerical solutions with much larger time steps than those required by explicit methods for maintaining numerical stability. As our system is also stiff -i.e. with time constants ranging various orders of magnitude- and the use of small time steps is nonetheless mandatory, using explicit methods would require extremely small time step sizes (~ps) resulting to the evaluation of an extremely large number of steps. Off course for real time co-simulations this would also call for equivalently small computational times. Even assuming the best case scenario of a single step calculation requiring just a single FPGA clock cycle, the minimum time step size would be limited by the FPGA clock frequency (typically up to 500MHz) to about 2ns. But even for off-line co-simulations the smaller computational cost per step offered by explicit methods will almost certainly be well overcompensated by the huge increase in number of steps, resulting thereby to prohibitive overall simulation times. It

should also be noted here that with the FPGAs calculation parallelism the difference in the computational cost per step between implicit and explicit methods is greatly nullified. Furthermore it should be taken into account that a round-off error is introduced in every time step because of the finite word length employed in the calculations. For example for 32bit/64bit floating point words (fig. 2) or equivalently fixed-point words with 23/52 bits in the fractional part, the accuracy is restricted at best to approximately 7/13decimal places. Therefore if the step size is extremely small, the small differences in the calculated numbers will most likely be rounded and since the number of steps is large, the accumulated round-off errors will have a significant impact on accuracy (and even on numerical stability).

Multi-step and single-step methods can both be used in our case provided they are stable with sufficiently high order (i.e. rate of decrease of the accumulated truncation error (global error) with the time step size). Single-step methods evaluate the value of a state at the next time step using the solution of the present time point and estimated values at points between the present and next time step (minor points). Multi-step methods, on the other hand, approximate the state derivatives by polynomials and thereby use many past state solutions for the next-time-step state-value calculation. Provided that the state trajectories are smooth the greater the number of points employed (either minor or past) the higher the achievable integration order. Therefore since single-step methods require evaluation of the state derivative functions at many minor points their computational cost per time step is generally higher than that of multi-step ones which use only past values that are already available. And since these minor point single-step calculations are primarily sequential, the use of FPGAs cannot nullify this burden. It should be mentioned, however, that multi-step methods are not well suited when the states exhibit discontinuities, since fitted polynomials based on past values are not valid in the neighbourhood of a discontinuity and their use in such case will lead to inaccuracies.

Lastly it should be highlighted that there is a fundamental trade-off between stability and accuracy properties with only second order methods being unconditionally stable - i.e producing bounded solution for any time step size (Hoffman 2001). Generally stability is a requirement in order to damp numerical oscillations and keep the solution bounded. However the initial amplitude of these oscillations depends on the order of the numerical integration method. In any case it would be undesirable to produce an under-damped but still oscillatory response because of large numerical errors. Also the existence of oscillations means that a small time step size is required to capture them making the method highly inefficient and computationally expensive. Therefore it is usually beneficial to trade some stability for higher order and there are some techniques which attempt to find the best compromise.

3. PHYSICS-BASED, ANALYTICAL PIN DIODE AND IGBT MODELS

This section first gives an overview of the basic geometric structure of the considered power electronic devices and then provides a detailed analytical solution and modelling of the relevant equations.

3.1. Basic Device Structure

The basic non-punch-through (NPT) structure of the considered devices is shown in figures 4(a) and 4(b) for the PIN diode and the IGBT respectively.



Figure 4 - Structure of NPT (a) PIN diode, (b) IGBT

The NPT PIN diode consists of a wide N- doped or intrinsic (nb denoted by I) drift region (nb also called base region) sandwiched between a thin P doped anode and a thin N⁺ doped cathode. The resulting P⁺N⁻ and N⁺N⁻ (or N⁺I) junctions are referred to as junctions J₁ and J₂ respectively. Similarly to diode at the one end, the NPT IGBT is featuring a P⁺ anode (collector) and a wide N⁻ drift region forming a P⁺N⁻ junction (junction J₁). However the IGBT is a gate controlled device with a MOSFET-like structure at the other end –i.e. P well, N⁺ emitter/source and MOS gate. The P well/N⁻ drift region and the N⁺ source/P well junctions are referred to as junctions J₂ and J₃ respectively.

3.2. Bipolar Device 1D Modeling

Generally circuit simulation works by solving a system of linear or linearized differential equations describing the circuit states (voltages/currents of each component). Main purpose of device modeling is to dynamically relate the voltage across the device with the current flowing though it so that the necessary values can be calculated at each time step.

For both the PIN Diode and the IGBT the device voltage can be considered as the voltage across the base; which in turn can be divided into the voltage across the carrier-storage-region (CSR), the junction (Boltzmann) voltages and the voltages across the space charge regions. The latter includes the depletion layers formed around reversed biased junctions and the nonconductivity modulated regions in the base (known as drift regions). In order to realistically model these two bipolar devices, the distributed nature of the charge storage in the drift region as well as the floating nature of the boarders of the CSR have to be accounted. Nonetheless, a dynamic solution is required only for the base region. All the other device parts can be considered as behaving quasi-statically; as suppliers/collectors of charge carriers to and from the drift region (Leturcq, P., 1997) or in mathematical terms as dynamic boundary conditions for the evaluation of the boarders' positions and the CSR charge concentration profile. Knowledge of the latter two enables the computation of the voltages across the CSR and the space charge regions. The following sub-sections describe in detail the analytical evaluation of all the required quantities.

3.2.1. Model of the Carrier Storage Region (CSR)

The charge profile and the depletion layers are primarily one-dimensional (nb for the diode for almost 100% while for the IGBT for over 90% of the drift region), thus a 1-D solution is adequate. Also as a convention in both devices a 1D base region width can be assumed, denoted by W_B . Under the conditions of quasi-neutrality and high levels of injection, the charge dynamics are described by the ambipolar diffusion equation (ADE), which in 1D is:

$$D\frac{\partial^2 p(x,t)}{\partial x^2} = \frac{p(x,t)}{\tau_{hl}} + \frac{\partial p(x,t)}{\partial t}$$
(1)

where D is the ambipolar diffusion coefficient, τ_{hl} is the high-injection-level carrier lifetime and p(x,t) is the ambipolar charge carrier density (nb excess holes (Δp) and excess electrons (Δn) in equal concentrations).

Since the carrier distribution inside the CSR can be written as a continuous function, the solution of (1) can be expressed as a cosine Fourier series (Leturcq 1997):

$$p(x,t) = v_{o}(t) + \sum_{k=1}^{\infty} v_{k}(t) \cos\left(\frac{k\pi(x-x_{1})}{x_{2}-x_{1}}\right)$$
(2)

where v_0 is the DC component of p(x, t) at time t, v_k is the amplitude of the kth harmonic and $x_1 \& x_2$ are the abscissae of the floating boarders of the charge storage region. Furthermore, the Fourier series coefficients v_k (for k=0, 1, 2, 3, ...) can be expressed as follows:

$$v_{k}(t) = \frac{1}{x_{2} - x_{1}} \int_{x_{1}}^{x_{2}} p(x, t) \cos\left(\frac{k\pi(x - x_{1})}{x_{2} - x_{1}}\right) dx$$
(4)

Multiplying (1) by the cosine term, then integrating it w.r.t. x from x_1 to x_2 while using (4), the ambipolar diffusion equation transforms into the infinite system of first-order differential equations described by (5):

$$\frac{\mathrm{d}v_{\mathrm{o}}(t)}{\mathrm{d}t} = -\frac{v_{\mathrm{o}}(t)}{\tau} + \frac{1}{(x_{2} - x_{1})} D\left(\frac{\partial p(x, t)}{\partial x}\Big|_{x_{2}} - \frac{\partial p(x, t)}{\partial x}\Big|_{x_{1}}\right)$$
$$-\frac{1}{(x_{2} - x_{1})} \sum_{n=1}^{\infty} v_{n}(t) \left[\frac{\mathrm{d}x_{1}}{\mathrm{d}t} - (-1)^{n} \frac{\mathrm{d}x_{2}}{\mathrm{d}t}\right] \quad (\text{for } k = 0)$$
(5)

$$\begin{aligned} \frac{\mathrm{d}v_{k}(t)}{\mathrm{d}t} &= \left(\frac{1}{\tau} + \frac{\mathrm{D}k^{2}\pi^{2}}{\left(x_{2} - x_{1}\right)^{2}} + \frac{1}{4}\frac{\mathrm{d}(x_{1} - x_{2})}{\mathrm{d}t}\right)v_{k}(t) \\ &+ \frac{2}{\left(x_{2} - x_{1}\right)}\mathrm{D}\cdot\left((-1)^{k}\frac{\partial p(x,t)}{\partial x}\Big|_{x_{2}} - \frac{\partial p(x,t)}{\partial x}\Big|_{x_{1}}\right) \\ &- \frac{2}{\left(x_{2} - x_{1}\right)}\sum_{\substack{n=1\\n \neq k}}^{\infty} \frac{n^{2}}{n^{2} - k^{2}}v_{n}(t)\left[\frac{\mathrm{d}x_{1}}{\mathrm{d}t} - (-1)^{n+k}\frac{\mathrm{d}x_{2}}{\mathrm{d}t}\right] \quad (\text{for } k \neq 0) \end{aligned}$$

The evaluation of the Fourier coefficients from system (5) requires the boundary conditions at the two ends of the CSR, namely the abscissae of the boarders $x = x_1$ and $x = x_2$, their time derivatives $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$, and the carrier concentration gradients at these points $g_1(t) = \frac{\partial p(x,t)}{\partial x}\Big|_{x_1}$ and $g_2(t) = \frac{\partial p(x,t)}{\partial x}\Big|_{x_2}$. The charge carrier distribution and the associated boundary

carrier distribution and the associated boundary conditions are illustrated in figure 5 for the case of static conduction and turn off transient (for the PIN diode fig. 5(a,b)) and for the IGBT fig. 5(c)).



Figure 5 - Charge carrier profile & boundary conditions for (a) Diode during static conduction (b) Diode during turn-off (c) IGBT during static conduction and turn-off

The gradients $g_1(t)$ and $g_2(t)$ can be calculated from the boundary hole and electron currents (i.e. I_{n1} , I_{p1} at $x=x_1$ and I_{n2} , I_{p2} at $x=x_2$) as in (6) where q is the electron charge, A is the junction area and D_n and D_p are respectively the electron and hole diffusion coefficients (Palmer et al. 2003). The coefficients D_n , D_p and D are calculated by equation (7) where μ_n and μ_p are the electron and hole mobilities, k is the Boltzmann constant and T is the junction temperature in K.

$$g_{1}(t) = \frac{1}{2qA} \left(\frac{I_{n1}}{D_{n}} - \frac{I_{p1}}{D_{p}} \right) \qquad g_{2}(t) = \frac{1}{2qA} \left(\frac{I_{n2}}{D_{n}} - \frac{I_{p2}}{D_{p}} \right) \quad (6)$$

$$D_p = \frac{kT}{q}\mu_p$$
 (i) $D_n = \frac{kT}{q}\mu_n$ (ii) $D = \frac{2D_nD_p}{D_n + D_p}$ (iii) (7)

3.2.2. Diode Boundary Conditions

For the diode, the emitter layers (i.e. the P⁺ anode and the N⁺ cathode layers) act as recombination sinks for the minority carriers. Thus the resulting minority currents I_{n1} , I_{p1} can be characterised by the conventional "h parameters" (Schlangenotto et al. 1969) and calculated using equation (8) where h_p is the recombination parameter at the P^+ layer, h_n is the recombination parameter at the N⁺ layer and p_{x1} & p_{x2} are respectively the excess carrier concentrations at $x=x_1$ and $x=x_2$. The majority currents I_{p1} , I_{n2} can then be calculated by equation (9) where I_A is the total anode current and Idisp1 & Idisp2 are the displacement currents charging respectively the capacitances of the depletion layers around junctions J_1 and J_2 (nb if applicable). The latter can be evaluated using equation (10) where ε is the material's electric permittivity, W_{d1} and W_{d2} are the widths of the respective depletion regions and V_{d1} and V_{d2} are respectively the voltages across them. W_{d1} and W_{d2} are related to V_{d1} and V_{d2} according to equation (11) which is classically derived from the Poisson equation. In (11) an effective base region doping concentration is considered which comprises of the background doping, N_B, and the density of free carriers moving though the region at their saturated velocity v_{sat}. The depletion layer voltages V_{d1} and V_{d2} are in turn calculated from the boundary carrier densities using equation (12) which is derived by applying proportional control with feedback (similar to op-amp circuits) from the boundary carrier densities p_{x1} & p_{x2} . The feedback is such that whenever the depletion layers are active, the concentrations p_{x1} & p_{x2} are kept equal to zero as physically expected. In control theory terms the values of p_{x1} & p_{x2} are equivalent to error signals and the motion of the boundaries x_1 and x_2 (via the change of the depletion region widths) is equivalent to the action for compensating that error. In our case the depletion layer widths are linked to p_{x1} and p_{x2} indirectly via the depletion layer voltages. It is possible to link them directly (proportionally) but this has been found causing convergence and accuracy issues (Bryant et al. 2007a). Finally, the abscissae of the CSR boarders can be found from the depletion layer widths as in (13).

Equations for the Diode boundary conditions:

$$I_{n1} = qAh_p(p_{x_1})^2$$
 $I_{p2} = qAh_n(p_{x_2})^2$ (8)

$$\begin{array}{cccc} \mathbf{1}_{p_1-\mathbf{1}A} - \mathbf{1}_{n_1} - \mathbf{1}_{disp_1} & \mathbf{1}_{n_2} - \mathbf{1}_A - \mathbf{1}_{p_2} - \mathbf{1}_{disp_2} & (9) \\ \mathbf{1}_{s_1} & \mathbf{1}_{s_2} & \mathbf{1}_{s_1} - \mathbf{1}_{s_1} & \mathbf{1}_{s_2} - \mathbf{1}_{s_1} - \mathbf{1}_{s_1} & \mathbf{1}_{s_1} - \mathbf{1}_{s_1} & \mathbf{1}_{s_2} - \mathbf{1}_{s_1} - \mathbf{1}_{s_1} & \mathbf{1}_{s_2} - \mathbf{1}_{s_1} & \mathbf{1}_{s_2} - \mathbf{1}_{s_1} & \mathbf{1}_{s_2} & \mathbf{1}_{s_1} & \mathbf{1}_{s_1} & \mathbf{1}_{s_2} & \mathbf{1}_{s_1} & \mathbf{1}_{s_1} & \mathbf{1}_{s_2} & \mathbf{1}_{s_1} & \mathbf{1}_{s_1} & \mathbf{1}_{s_1} & \mathbf{1}_{s_1} & \mathbf{1}_{s_1} & \mathbf{1}_{s_2} & \mathbf{1}_{s_1} & \mathbf{1}_{$$

$$I_{disp1} = \frac{u}{W_{d1}} \frac{u}{dt} \qquad I_{disp2} = \frac{u}{W_{d2}} \frac{u}{dt}$$
(10)

$$W_{d1} = \sqrt{\frac{2\varepsilon V_{d1}}{qN_{B} + \frac{|I_{P1}|}{AV_{sat}}}} \qquad W_{d2} = \sqrt{\frac{2\varepsilon V_{d1}}{qN_{B} + \frac{|I_{n2}|}{AV_{sat}}}}$$
(11)

$$V_{d1} = \begin{cases} 0 & p_{x_1} \ge 0 \\ -K_F p_{x_1} & p_{x_1} < 0 \end{cases} \quad V_{d2} = \begin{cases} 0 & p_{x_2} \ge 0 \\ -K_F p_{x_2} & p_{x_2} < 0 \end{cases}$$
(12)

$$x_1 = W_{d1}$$
 $x_2 = W_B - W_{d2}$ (13)

3.2.3. IGBT Boundary Conditions

The boundary conditions for the NPT IGBT are identical to that of the diode at the end $x=x_1$ with the P⁺ IGBT collector layer serving as a recombination sink for the electrons. Therefore, I_{n1} and I_{p1} can be found by equation (14) with I_C being the collector current and the rest of the symbols having the same meaning as before. It should be noted here that there is no displacement current I_{disp1} since the junction J₁ is always forward biased and thereby no depletion layer is present there. This also means that the boundary x_1 is always fixed at x=0. At $x=x_2$ the situation is different from the diode with I_{n2} now being the MOS channel electron current, and the I_{p2} the hole current collected by the P-well. The value of In2 can be calculated by the typical MOSFET equations (15(i)-(16)) where V_{GE} is the voltage applied to the gate (G) w.r.t the IGBT N^+ source/emitter (E), V_{TH} is the threshold voltage for channel inversion, V_{DE} is the voltage between the "virtual drain" (D) and the emitter, K_p is the MOS transconductance and λ is the channel shortening parameter. The value of Ip2 can then be found from (15(ii)) i.e. by subtracting I_{n2} and the two displacement currents at $x=x_2$ from I_C. The latternamely I_{disp2} and I_{GC} - represent respectively the current charging the collector-emitter capacitance, C_{CE} (formed by the depletion region under the p-well) and the current charging the collector-gate (Miller) capacitance, C_{CG} (nb. or equivalently collector- "virtual-drain" capacitance). The values of I_{disp2} and I_{CG} are calculated from (17), (18) where C_{ox} is the oxide capacitance per unit area, α_i is the ratio of the intercell area (nb. area between P-wells) to the total die area, A is the total die area, l_m is the half intercell width and W_{d2} is the depletion region width formed under the P-wells. The latter is associated with the voltage across the layer, V_{d2} , as in the case of the diode via equation (19(i)) with V_{d2} again serving as the control actuator (equation (20)) for keeping the concentration p_{x2} equal to zero (nb in this case the time). Equations (17)-(18) can be derived respectively from $I_{disp2} = C_{CE} \frac{dV_{DE}}{dt} \& I_{CG} = C_{CG} \frac{dV_{DG}}{dt}$ and the expressions of the relevant capacitances (Palmer

and the expressions of the relevant capacitances (Palmer et al. 2003). It should be noted here that the value of V_{DE} is assumed to be equal to V_{d2} since typically the potential at the edge of the depletion layer around the P- well is approximately equal to that at the end of the MOS channel. Lastly, x_2 is found by equation (19(ii)).

Equations for the IGBT boundary conditions:

$$I_{p1} = I_c - I_{n1}$$
 $I_{n1} = qAh_p (p_{x_1})^2$ (14)

$$I_{p2} = I_{A} - I_{n2} - I_{disp2} - I_{GC} (i) \quad I_{n2} = \begin{cases} I_{MOS} \text{ if } V_{GE} \ge V_{TH} \\ 0 \text{ if } V_{GE} < V_{TH} \end{cases} (ii)$$
(15)

$$I_{MOS} = \frac{K_p}{2} \left[2V_{DE}(V_{GE} - V_{TH}) - V_{DS}^2 \right] \text{ if } V_{DE} \leq V_{GE} - V_{TH} \quad (16)$$

or
$$= \frac{K_p}{2} \left[(V_{GE} - V_{TH})^2 \left(1 + \lambda (V_{DE} - (V_{GE} - V_{TH})) \right) \right] \text{ otherwise}$$

$$I_{disp2} = \frac{\varepsilon A(1 - \alpha_i)}{W_{d2}} \frac{dV_{d2}}{dt}$$
(17)

$$I_{CG} = \frac{C_{ox}A\alpha_{i}}{1 + \frac{C_{ox}}{\varepsilon} \left(W_{d2} - l_{m} - \sqrt{\frac{2\varepsilon V_{GE}}{qN_{B}}}\right)} \left(\frac{dV_{d2}}{dt} - \frac{dV_{GE}}{dt}\right)$$
(18)

$$W_{d2} = \sqrt{\frac{2\varepsilon V_{d1}}{qN_{B} + \frac{|I_{C}|}{Av_{sat}}}} (i) \qquad x_{2} = W_{B} - W_{d2} (ii) \quad x_{1} = 0 (iii) \quad (19)$$
$$V_{d2} = \begin{cases} 0 \quad p_{x_{2}} \ge 0 \\ -K_{F} p_{x_{2}} \quad p_{x_{2}} < 0 \end{cases} \quad (20)$$

3.2.4. Charge Storage Region (CSR) Voltage Drop

The voltage drop across the quasi-neutral CSR region, V_B , can be found by calculating the integral expression (21) where J is the total device current density. In physical terms the first integral represents a purely ohmic voltage drop, V_{Ω} , whilst the second one is the Dember voltage, V_{Dember} , which is much smaller than V_{Ω} and is therefore safely neglected.

$$V_{\rm B} = V_{\Omega} + V_{\rm dember} = \frac{J}{q(\mu_{\rm n} + \mu_{\rm p})} \int_{x_{\rm l}}^{x_{\rm 2}} \frac{dx}{p_{\rm x}} - \frac{kT}{q} \left(\frac{\mu_{\rm n} - \mu_{\rm p}}{\mu_{\rm n} + \mu_{\rm p}}\right) \int_{p(x_{\rm 2})}^{p(x_{\rm 1})} \frac{dp}{p} (21)$$

where $p_x = p_T(x) = p(x) + \mu_n N_B / (\mu_n + \mu_p)$ for a time t

Since an analytical evaluation of V_{Ω} is impossible with p_x expressed in Fourier series, a piecewise linear approximation of p_x is instead considered. More specifically the concentration is evaluated (using the Fourier series) at M+1 equidistant points and then assumed to vary linearly between them as in figure 6.



Figure 6- Discretisation of charge carrier profile in CSR

The total base voltage can then be calculated by (22):

$$V_{B} = \frac{J(x_{2} - x_{1}) / M}{q(\mu_{n} + \mu_{p})} \times \sum_{m=1}^{M} \left(\gamma \frac{\ln \left[\left| \frac{N_{B} + p(x_{lm})}{N_{B} + p(x_{lm-1})} \right| \right]}{p(x_{lm}) - p(x_{lm-1})} - (\gamma - 1) \frac{1}{p_{x_{lm}}} \right)$$

where
$$\gamma = \begin{cases} 1 \text{ if } p(x_m) \neq p(x_{m-1}) \\ 0 \text{ if } p(x_m) = p(x_{m-1}) \end{cases}$$
 and $x_{1m} = x_1 + m(x_2 - x_1) / M$

Note: p_{x1} and p_{x2} are respectively equivalent of p_{x10} and p_{x1M} . Similarly for $p(x_1)$ and $p(x_2)$

It should be noted that for the case of the IGBT the assumption of a purely 1D profile needs to be treated carefully at the MOS end of the device. As described previously in the evaluation of the boarder position, x_2 , the model assumes an effective depletion region width equal to that formed under the p-well and controlling p_{x_2} to be equal to zero. However for the calculation of the ohmic drop, V_{Ω} , the effect of the accumulation charge under the gate overlap should be taken into account as otherwise an unrealistically high resistance will be predicted. To account for this effect the value of $p(x_2)$ used in equation (22) is derived from the penultimate one according to $p(x_2) \approx \alpha_i p(x_{M-1})$.

3.2.5. Calculation of the Devices' Terminal Voltages

For the diode the terminal anode-cathode voltage, V_{AK} is given by equation (23), i.e. it is equal to the sum of the following voltages: (i) the two Boltzmann junction voltages V_{J1} and V_{J2} (equation 24), (ii) the two space charge region voltages V_{d1} and V_{d2} (equation 12) and (iii) the CSR voltage, V_B , (equation 22 where J=I_A/A).

$$V_{AK} = V_{J1} + V_{J2} + V_{d1} + V_{d2} + V_B$$
(23)

$$\mathbf{V}_{J1} = \frac{kT}{q} \ln \left(\frac{\mathbf{p}_{x_1} \mathbf{N}_B}{\mathbf{n}_i^2} \right) \qquad \qquad \mathbf{V}_{J2} = \frac{kT}{q} \ln \left(\frac{\mathbf{p}_{x_2}}{\mathbf{N}_B} \right)$$
(24)

For the IGBT the terminal collector-emitter voltage, V_{CE} is given by equation (25), i.e. it is equal to the sum of the following voltages: (i) the Boltzmann junction voltage V_{J1} (equation 24), (ii) the space charge region voltage V_{d2} (equation 20) and (iii) the CSR voltage, V_B , (equation 22 where $J=I_C/A$).

$$V_{CE} = V_{J1} + V_{d2} + V_B$$
(25)

The terminal gate–emitter voltage, V_{GE} , can be found by solving the differential equation (26) where I_G is the gate current, C_{GE} is the gate-emitter capacitance (constant) and C_{GC} is the non-differential part of equation (18). The differential equation (26) can be derived by considering Kirchhoff's current law (KCL) at the gate i.e. $I_{GE} = I_G + I_{CG}$.

$$\frac{\mathrm{d}V_{\mathrm{GE}}}{\mathrm{d}t} = \frac{1}{\mathrm{C}_{\mathrm{GE}} + \mathrm{C}_{\mathrm{GC}}} \left(\mathrm{I}_{\mathrm{G}} + \mathrm{C}_{\mathrm{GC}} \frac{\mathrm{d}V_{\mathrm{d}2}}{\mathrm{d}t} \right) \tag{26}$$

3.2.6. Temperature Dependency

The models include temperature effects through the temperature dependency of the various physical

parameters. The required device temperature can be calculated via a thermal model (e.g. RC network) of the device packaging. More details about temperature dependency can be found in (Palmer et al. 2003).

4. STRATEGY FOR FPGA IMPLENTATION

4.1. Co-simulation Method

As described in section 2.3.2 when simulating stiff systems like power electronic circuits, an implicit method should be preferred due to its stability. However, a true implicit approach would call either for the system states to be processed all in one platform or in the case of co-simulation it would call for a mixed, concurrent processor/FPGA evaluation of the involved matrices ultimately leading to a very time-consuming simulation due to the huge number of (different) data exchanges and thereby communication overheads during a single time step. The solution to the problem is system partitioning i.e. decoupling of the system to at least two portions, one or more for processing in FPGA and one or more for processing in software.



Figure 7 – Block diagram representation of system partitioning into two subsystems

Without any loss of generality figure 7 shows a functional diagram of the considered method where the circuit is divided into two subsystems passing to each other the required coupling state variables $y_1 \& y_2$ augmented by unavoidable errors ε_1 and ε_2 (due to sampling as well as due to truncation and rounding by the selected integration method) into inputs u_1 and u_2 . Once the system is decoupled, different integration methods and/or time step sizes can be used in order to ensure stability (Gear et al. 1984). Furthermore, because of partitioning the matrices are less sparse resulting to only few unnecessary zero-element multiplications and hence more efficient usage of the hardware resources.

For illustration purposes the generalised chopper circuit of figure 8 is chosen to be modeled. Although it is possible to split the system by pure numerical considerations, here for simplicity the system is partitioned naturally. In particular the system is separated into four subsystems: (i) the PIN diode model with internal states \mathbf{x}_{diod} as defined by equation (5), (ii) the IGBT model with internal states \mathbf{x}_{igbt} as defined by equations (5) and (26), (iii) the main circuit with states \mathbf{x}_{circ} and (iv) the gate drive circuit with states \mathbf{x}_{driv} . The first two subsystems are chosen to be implemented in FPGA (since they are the most computationally intensive) while the other two in software.



Figure 8 - The generalised chopper cell circuit

With the considered approach the entire system with states \mathbf{x} can be decomposed as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{x}}_{diod} \\ \dot{\mathbf{x}}_{igbt} \\ \dot{\mathbf{x}}_{circ} \\ \dot{\mathbf{x}}_{driv} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{diod} \left(\mathbf{x}_{diod}, \mathbf{w}_{diod}, t \right) \\ \mathbf{f}_{igbt} \left(\mathbf{x}_{igbt}, \mathbf{w}_{igbt}, t \right) \\ \mathbf{f}_{circ} \left(\mathbf{x}_{circ}, \mathbf{w}_{circ}, t \right) \\ \mathbf{f}_{driv} \left(\mathbf{x}_{driv}, \mathbf{w}_{driv}, t \right) \end{bmatrix}$$
(27)

where \mathbf{f}_{diod} , \mathbf{f}_{igbt} , \mathbf{f}_{circ} , \mathbf{f}_{driv} and \mathbf{w}_{diod} , \mathbf{w}_{igbt} , \mathbf{w}_{circ} , \mathbf{w}_{driv} are respectively the vector functions and vector inputs of the diode, IGBT, main circuit and gate drive circuit subsystems including the coupling variable(s) passed into them. More specifically: $\mathbf{w}_{diod} = \mathbf{y}_{circ_1}$, $\mathbf{w}_{driv} = \mathbf{y}_{igbt_2}$, $\mathbf{w}_{igbt} = \begin{bmatrix} \mathbf{y}_{circ_2} \cdot \mathbf{y}_{driv} \end{bmatrix}^T$, $\mathbf{w}_{circ} = \begin{bmatrix} V_{DC} \cdot \mathbf{y}_{igbt_1} \cdot \mathbf{y}_{diode} \cdot \mathbf{y}_{driv} \cdot \mathbf{y}_{circ_2} \end{bmatrix}^T$ with the coupling variables being $\mathbf{y}_{diode} = V_{AK}$, $\mathbf{y}_{igbt_1} = V_{CE}$, $\mathbf{y}_{igbt_2} = V_{GE}$, $\mathbf{y}_{circ_1} = I_A$, $\mathbf{y}_{circ_2} = I_C$, $\mathbf{y}_{driv} = I_G$

It should be noted here that to the internal states of the devices' subsystems it is required to add extra states for the calculation of the derivatives as required by system (5) and its internal functions (10), (17) and (18).

4.2. Numerical Integration

The decomposed states of the several subsystems are calculated in parallel using a suitable numerical integration method. As explained in section 2.3.2 the preferred methods should be (i) fixed time step-size, (ii) implicit and (iii) multistep. For such methods the general solution equation for the value of the next-time-step subsystems' states can be written as in (28)

$$\mathbf{x}_{sub}^{\mathbf{n}+1} = \sum_{i=0}^{p} \alpha_{i} \cdot \mathbf{x}_{sub}^{\mathbf{n}\cdot\mathbf{i}} + \Delta t_{sub} \sum_{i=-1}^{p} \beta_{i} \cdot \mathbf{f}_{sub}^{\mathbf{n}\cdot\mathbf{i}}$$
(28)

where \mathbf{x}_{sub}^{m} and $\mathbf{f}_{sub}^{m} = \mathbf{f}_{sub}(\mathbf{x}_{sub}^{m}, \mathbf{w}_{sub}^{m}, t^{m})$ are respectively the state and function vectors of the subsystem at the \mathbf{m}^{th} time step, α_{i} and β_{i} are method-dependent polynomial coefficients (with $\beta_i \neq 0$ for implicit methods), and Δt_{sub} is the subsystem's integration timestep size. Based on stability, efficiency, accuracy and real-time constraints considerations the chosen method for the devices' subsystems is the 3rd order Gear Method (Gear 1971) while that for the circuit subsystems is the 3rd order Brayton method (Brayton 1972) with some modifications as proposed by Shampine (Shampine 1980) and others.

In order to solve equation (28) explicitly in terms of \mathbf{x}_{sub}^{n+1} , the vector function \mathbf{f}_{sub} must be linear. For the cases of the diode and the IGBT subsystems where \mathbf{f}_{sub} is non-linear, this should be linearized according to:

$$\mathbf{f}_{sub} = \frac{\partial \mathbf{f}_{sub}}{\partial \mathbf{x}_{sub}} \mathbf{x}_{sub} + \frac{\partial \mathbf{f}_{sub}}{\partial \mathbf{w}_{sub}} \mathbf{w}_{sub} = \mathbf{A}_{sub} \mathbf{x}_{sub} + \mathbf{B}_{sub} \mathbf{w}_{sub}$$
(29)

Then the evaluation of the time advanced states can be done as in (30):

$$\mathbf{x}_{sub}^{n+1} = \Delta t_{sub} \left(\mathbf{I} - \Delta t_{sub} \beta_{\cdot 1} \mathbf{A}_{sub}^{n+1} \right)^{-1} \\ \times \left[\beta_{\cdot 1} \mathbf{B}_{sub}^{n+1} \mathbf{w}_{sub}^{n+i} + \sum_{i=0}^{p} \left(\alpha_{i} \mathbf{x}_{sub}^{n-i} + \beta_{i} \mathbf{f}_{sub}^{n-i} \right) \right]$$
(30)

where \mathbf{f}_{sub}^{n+1} has been substituted by its linear/linearized form as in (29) with \mathbf{A}_{sub}^{n+1} , \mathbf{B}_{sub}^{n+1} being the Jacobian matrices of the next time step (nb these are calculated numerically using the small perturbation theory). The inverse of the matrix $\mathbf{M} = (\mathbf{I} - \Delta t_{sub} \beta_{-1} \mathbf{A}_{sub}^{n+1})$ is found using LU decomposition followed by forward and backward substitution.

4.3. FPGA Parallel Programming

In FPGAs the matrix multiplications involved in equation (30) can all be performed in parallel in two steps as a sum of products (SOP) resulting to great computational speed ups compared to serial processing. For example for a matrix-matrix MN multiplication, where **M** and **N** are mxm and mxn matrices, the computational cost reduces from mn(2m-1) sequential operations to m²n parallel multiplications and m(m-1)n parallel additions. Besides from the number of calculation steps the simulation speeds also depend on the overheads imposed by the several processing equipment and communication links. Since the most critical overhead is that of software-hardware data transactions, the considered methodology reduces the number of software-FPGA interactions via the use of frame signals, i.e. data transaction parallelism. In particular a series of input/output data values are stored in a data vector which is transferred in one burst into/from an input/output buffer residing on the FPGA.

4.4. Practical Implementation

The targeted hardware platform is a Digilent Atlys development board with Xilinx Spartan 6 FPGA. The co-simulation environment is set up in Simulink with the software-hardware interface being provided by Xilinx System Generator (XSG), (Xilinx Inc., 2012c), and the communication between the PC and the board done via 1Gbit Ethernet. An important advantage of this method is that it enables rapid implementation and verification of the algorithms employed as opposed to the more classical approach. The entire system development requires the following steps:

- 1. Floating-point algorithm development and verification in Matlab/Simulink via the use embedded Matlab functions.
- 2. Design elaboration for FPGA implementation by including the use of fixed point arithmetic and other hardware constraints. The former requires selection of the appropriate bit widths and radix point position for each block in order to enhance/maximise the signal to noise ratio (SNR). A safety margin of four bits is also added to avoid overflows. It should be noted here that the several state variables require appropriate scaling in order to reduce the effect of round off errors and increase precision.
- 3. Design adjustment for bottleneck removal and efficient exploitation of the hardware resources via critical paths' pipelining and time division multiplexing for resource sharing of unutilized functional blocks.
- Generation/writing of HDL (VHDL/Verilog) code for the several hardware subsystems (nb only the device models for the circuit of figure 8). Placement of the relevant HDL codes into XSG "black boxes".
- 5. Set up of the software-hardware interfacing via Xilinx Gateway blocks as well as FIFO buffers and control for frame-based processing. For the latter the FPGA is used in free run mode.
- 6. Automatic generation of BIN file using the System Generator with Xilinx ISE.

5. CONCLUSIONS

In this paper the strategy for the FPGA implementation of analytical power device models incorporated in a circuit has been described. A detailed comparison of the co-simulation results and the achieved computational accelerations will be presented in a later publication. Further work is required towards the development of a multi-FPGA platform in order to achieve hardware implementation of larger circuit systems employing multiple power devices. Furthermore apart from cosimulation purposes the device models will also be exploited for real-time model-based control.

REFERENCES

- Altera Corp., 2012. Cyclone IV Device Handbook. Available from: http://www.altera.com/literature/hb/cyclone-iv/ cyclone4-handbook.pdf [accessed 12 July 2012]
- Altera Corp., 2012. *Stratix IV Device Handbook*. Available from: http://www.altera.com/literature/hb/stratix-iv/stratix4_ handbook.pdf [accessed 12 July 2012]
- Brayton, R.K., Gustavson, F.G., Hachtel, G.D., 1972. A new Efficient Algorithm for Solving Differential-Algebraic Systems Using Implicit Backward Differentiation

Formulas, *IEEE Transactions on Industry Applications*, 60 (1): 98–108.

- Bryant, A.T., Palmer, P. R., Santi, E., and Hudgins, J.L., 2007. Simulation and optimization of diode and Insulated Gate Bipolar Transistor interaction in a chopper cell using MATLAB and Simulink, *IEEE Transactions on Industry Applications*, 43 (4): 874–883.
- Bryant, A.T., Santi, E., Palmer, P. R., and Hudgins, J.L., 2007. Two-step parameter extraction procedure with formal optimization for physics-based circuit simulator IGBT and p-i-n Diode Models, *IEEE Transactions on Power Electronics*, 21 (2): 295–309.
- Hoffman, J.D., 2001. Numerical Methods for Engineers and Scientists. 2nd Ed. New York: Marcel Dekker Inc.
- Gear, C.W., 1971. Simultaneous Numerical Solution of Differential-Algebraic Equations, *IEEE Transactions on Circuit Theory*, 18 (1): 89-95
- Gear, C.W. and Wells, D.R., 1984. Multirate linear multistep methods, *BIT Numerical Mathematics*, 24 (4): 484–502.
- Gonzalez J. and Nunez R., 2009. LAPACKrc: Fast linear algebra kernels/solvers for FPGA accelerators. *Journal of Physics: Conference Series, SciDAC 2009*, IOP Publishing, 180012042.
- Greenberg, M.D., 1998. Advanced Engineering Mathematics. 2nd Ed. New Jersey: Prentice Hall.
- IEEE, 2008. 754-2008 IEEE Standard for Floating-Point Arithmetic. Available from: http://ieeexplore. ieee.org [accessed 12 July 2012]
- Jin, H., 1997. Behavior-Mode Simulation of Power Electronic Circuits, *IEEE Transactions on Power Electronics*, 12 (3): 443-452.
- Leturcq, P., 1997. A study of distributed switching processes in IGBTs and other power bipolar devices Circuit simulator models for the diode and IGBT with full temperature dependent features, *Proceedings of Power Electronics Specialists Conference*, pp. 139-147. June 22-27, St. Louis (Missouri, USA).
- Oh, H.-S, Nokali M. El., 2001. A new IGBT behavioural model, Solid-State Electronics, 45 (12): 2069-2075.
- Palmer, P. R., Santi, E., Hudgins, J. L., Kang, X., Joyce, J. C. and Eng, P. Y., 2003. Circuit simulator models for the diode and IGBT with full temperature dependent features, *IEEE Transactions on Power Electronics*, 18 (5): 1220–1229.
- Pejovic, P. and Maksimovic, R., 1994. A Method for Fast Time-Domain Simulation of Networks with Switches, *IEEE Transactions on Power Electronics*, 9 (4): 449-456.
- Schlangenotto, H. and Gerlach W., 1969. On the effective carrier lifetime in p-s-n rectifiers at high injection levels, *Solid State Electronics*, 12: 267-275.
- Shampine, L.F., 1980. Implementation of implicit formulas for the solution of ODE's, *SIAM Journal on Scientific and Statistical Computing*, 1(1): 103-118.
- Strunz, K., 2004. Flexible Numerical Integration for Efficient Representation of Switching in Real Time Electromagnetic Transients Simulation, *IEEE Transactions on Power Delivery*, 19 (3): 1276-1283.
- Xilinx Inc., 2009. Hardware Co-Simulation Basics using System Generator for DSP. Available from: Xilinx webinars [accessed 12 July 2012]
- Xilinx Inc., 2012. *DS150*, *Virtex-6 Family Overview*. Available from: http://www.xilinx.com/support/ documentation/ data _sheets/ds150.pdf [accessed 12 July 2012]
- Xilinx Inc., 2012. DS160, Spartan-6 Family Overview. Available from: http://www.xilinx.com/support/documentation/data _sheets/ds160.pdf [accessed 12 July 2012]
- Xilinx Inc., 2012. System Generator for DSP. Available from: http://www.xilinx.com/support/documentation/sw_manuals /xilinx13_4/sysgen_user.pdf_[accessed 12 July 2012]
- Zack S. and Dhanani S., 2004. DSP Co-Processing in FPGAs: Embedded High-Performance, Low-Cost DSP Functions. *Xilinx White Paper*, ver.1.0, (March): 1–8.