Accelerated simulation of passive analogue VLSI interconnect on GPUs

Tom J. Kazmierski and Gines Domenech-Asensi

Abstract—Analogue models of VLSI interconnect in complex digital systems pose significant design challenges due to their size, often exceeding thousands of discrete nodes, and their tightly coupled structure. Such models need to be formulated and solved in the analogue domain to ensure high accuracy of interconnect effects such as signal delays and signal-to-signal interference. Currently available design tools are inadequate for simulating such systems due to prohibitive CPU times. This paper presents a technique which takes an advantage of the passive nature of interconnect such that simulations of large interconnect systems can be accelerated by about an order of magnitude compared with equivalent SPICE simulations. The acceleration is possible due to the use of explicit integration of the interconnect state equations were a fast estimate of the maximum allowed step-size is used to guarantee numerical stability. We show that a parallel implementation of the proposed algorithm is straightforward on GPU architectures and compare the results with those obtained from both standard and GPU implementations of SPICE. Several case studies are presented to illustrate the speed of the proposed method and to show that a good match can be obtained between the accuracy of standard SPICE-like simulations and the proposed approach. The proposed method has already been tested in RC interconnect simulations but here, for the first time, we present the performance of our method when applied to RLC interconnect which poses significant restrictions on the maximum allowed simulation step-size necessary to maintain numerical stability,

Index Terms—Simulation acceleration, state-space technique, many-core computer, GPU.

I. Introduction

Classical SPICE-like simulators used to analyse the behaviour of analogue circuits likerely on the modified nodal analysis and use implicit integration techniques based on the Newton-Raphson linearisation method to solve the circuit analogue equations at each time step. These methods have proven to be reliable and numerically stable, but on the other hand, they lead to long CPU times, often hours or even days and weeks for large circuits. These long simulation times contribute to delays in the design cycle time. The main reason for extensive computation times is due the necessity to build and factorise the Jacobian matrix of the analogue system multiple times at each time step. In contrast to implicit integration methods, the computational workload of explicit integration techniques is lighter. The main disadvantage of explicit methods is the need to limit the step size to ensure numerical stability. In a general case of a non-linear analogue system, where equations are stiff due to the large disparity of time constants, the step-size limitation may be very severe and in such cases implicit methods perform better. However, in the case of VLSI interconnect, equations are not stiff and what is more, interconnect is passive and usually linear which simplifies the state equation formulation for explicit methods, and makes estimates of the maximum allowed step-size easier and therefore faster. Different works have proved that the use of state-space equations combined with explicit integration methods is a suitable technique to speed up transient simulations of many types of analogue circuits [1] or mixed systems [2]. However, given the increasing complexity of analogue circuits and systems, new techniques are required to speed transients simulations, besides the use of alternative integration algorithms. Among these techniques, those based on exploiting the parallelisation of analogue integration methods running on parallel computer architectures are gaining more and more attention in the recent years. The Compute Unified Device Architecture (CUDA) [3] proposed by NVIDIA in 2006, is a programming model that allows engineers to use a high level programming language such as C to develop algorithms for general purpose Graphics Processing Units (GPUs). This has provided design engineers with software tools to use relatively cheap parallel architecture computers which now can perform fast simulations in different types of scientific and engineering applications. Thus, in the last decade, there have been many proposals to accelerate the simulation of analog circuits using GPUs e.g. [4]–[7]. Some works have focused on sparse matrix solvers [8] or LU factorization matrix solver [9]–[11], which have achieved substantial speedups compared with traditional parallel sparse solvers like PARDISO [12] or KLU [13]. NVIDIA also released an official sparse matrix solver, cuSolver [14], but the LU factorization in it is still performed on the CPU rather than GPU.

A common characteristic of these works is that they are focused on the traditional implicit integration methods used in SPICE-like simulators like. Recently, an explicit integration method parallelisable over a many-core processors has been proposed [15]. This method combines space state equations with a fixed-step explicit scheme based on the Adams-Bashforth integration formula to speed up the simulation of passive circuits of a complexity up to 1000 nodes.

In this paper we explore the method proposed in [15] further. We improve our fast estimate of the maximum allowed step size and we apply the technique to the RLC interconnect which poses a harder challenge on the step-size requirements than the RC interconnect considered in [15] . We show that in the case of RLC interconnect our method is not only faster than the

T.J. Kazmierski is with the Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom e-mail: tjk@ecs.soton.ac.uk

G. Domenech-Asensi is with the Departmento de Electronica y Tec. de Computadoras, Universidad Politecnica de Cartagena, Cartagena, 30201, Spain e-mail: gines.domenech@upct.es

parallel CU-SPICE by up to one order of magnitude for large systems but it is also significantly more accurate as it avoids spurious numerical ringing which characterises the implicit trapezoidal method used as the default integration scheme in CU-SPICE.

II. Stability analysis

Let (1) be the general state equation of a nonlinear, passive dynamic system:

$$\dot{x}(t) = f(x_t, t); x(0) = x_0 \tag{1}$$

As interconnect models are usually linear, the state equation at time point $t_k, k = 0, 1 \ldots$ can be formulated as:

$$\dot{X}(t_k) = A_k X(t_k) + E e_x \tag{2}$$

where X the vector of N state variable wave-forms, e_x a vector of excitations and A_k and E are coefficient matrices, where A_k is the Jacobian of the linearised model at the time point t_k. As the state equation 1 represents a passive system, the eigenvalues of the Jacobian A_k are guaranteed to have negative real parts, so explicit methods can be applied easily to provide a fast integration process. The step-size in explicit integration must be limited to ensure stability besides controlling the accuracy of the numerical solution. The computation of the maximum allowed step-size requires the computation of the spectral radius of $||A||$, a process for which time-consuming operations such as matrix multiplications and eigenvalue calculations are performed. In this work we take advantage of a recently developed fast method to calculate approximate step-size bounds for stability [2]. Although step sizes obtained using such approximate techniques are expected to be smaller than the maximum allowed step sizes calculated from the exact values of the Jacobians eigenvalues, the advantage of using approximate estimates is speed.

The stability of fixed-step Adams-Bashforth methods is defined by the well known stability plots shown in figure 1, where the values of maximum λh which guarantee stability are plotted in the complex plane. So, while for a first order method the maximum acceptable absolute value of λh is 2, for a fourth order method it is decreased to only 0.3. Thus, although higher order methods are more accurate than lower order ones, they are also more unstable.

Stability becomes harder to achieve when variable-step integration is used. Figure 2 shows a finite-difference grid for a q-order Adams-Bashforth method, where t_k is the current time point, t_{k+1} is the next time point, $P_q(t)$ is the interpolation polynomial of order q, and Δx is the unknown in the integration problem.

Let $h_i = t_{i+1} - t_i$ be the time step between two consecutive time points t_i and t_{i+1}. In a fixed-step integration method, all the h_i values are equal and invariable in time. However, in a variable-step method, the values of h_i are different, and change with time. The expressions for the general variable-step method can be obtained by integrating the divided difference polynomial approximation between the current variable value $x_k \equiv x(t_k)$ and the predicted one $x_{k+1} \equiv x(t_{k+1})$.

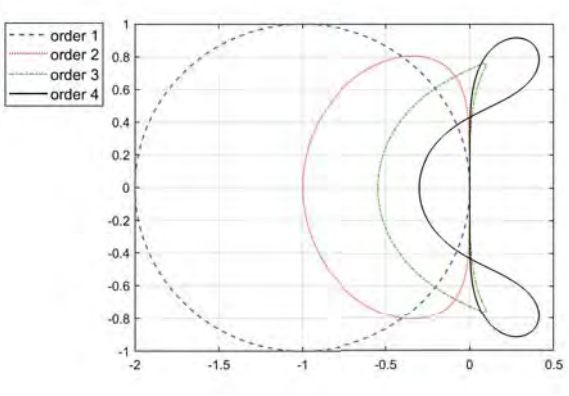

Fig. 1. Stability regions for Adams-Bashforth methods of order 1 to 4.

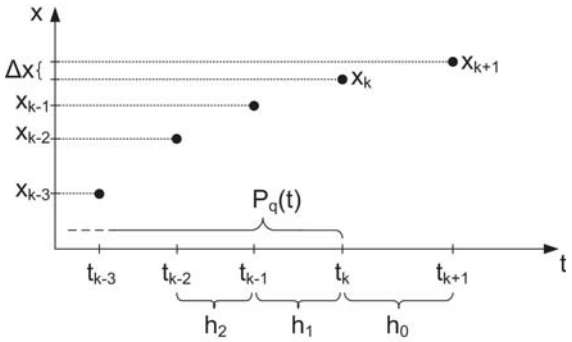

Fig. 2. Finite difference grid for the q^{th}-order Adams-Bashforth method.

$$\begin{aligned} \Delta x = \int_{x_k}^{x_{k+1}} dy = \int_{t_k}^{t_{k+1}} [P_q(t)]_k dt \Rightarrow \\ \Rightarrow x_{k+1} - x_k = \int_{t_k}^{t_{k+1}} \Big(f_0 + (x - x_k) f_k^{(1)} + \ldots \\ \ldots + (x - x_k)\ldots(x - x_{k-p}) f_k^{(q)} \Big) dt + O \end{aligned} \tag{3}$$

where $f_k^{(q)}$ is the q^{th} divided difference of function f at t_k [18] and O is the truncation error. Taking as example the second order method, the state variable at time t_{k+1} is computed as:

$$x_{k+1} - x_k = f_k h_0 \left(1 + \frac{h_0}{2h_1}\right) - f_{k-1} h_0 \left(\frac{h_0}{2h_1}\right) \tag{4}$$

being $f_k = f_k^{(0)}$. For the third order method, the following term is added to (4):

$$\begin{aligned} \Bigg(\frac{t_{k+1}^3 - t_k^3}{3} - (t_k + t_{k-1}) \frac{t_{k+1}^2 - t_k^2}{2} \\ + t_k t_{k-1} (t_{k+1} - t_k) \Bigg) f_k^{(2)} \end{aligned} \tag{5}$$

Figure 3 shows the stability plots for equations (4) and (5) integrated with different step-size expansion ratios from 1 to 2. The plots show how the variation of the step-size expansion ratio $r = h_{i+1}/h_i$ affects the stability for both the second and

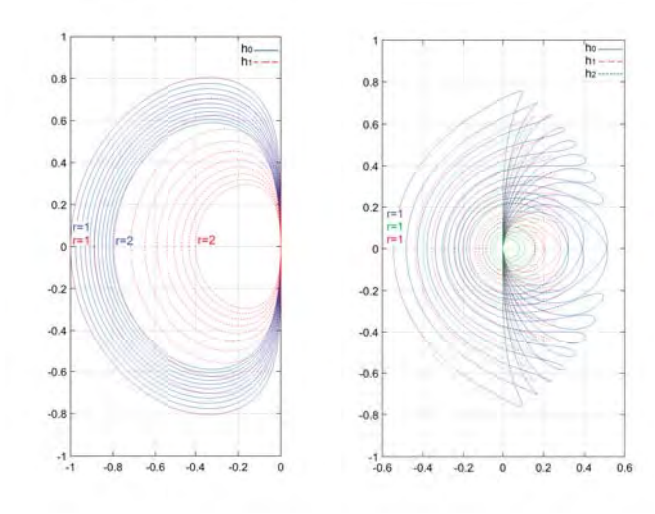

Fig. 3. Stability regions for the second and third order AB methods.

TABLE I
MAXIMUM REAL NEGATIVE VALUE OF THE STABILITY PLOTS

r	1	1.2	1.4	1.6	1.8	2
$L_r(h_0)$	-1	-0.952	-0.909	-0.869	-0.833	-0.8
$L_r(h_1)$	-1	-0.794	-0.649	-0.544	-0.463	-0.4

the third order AB methods. The values of the integration step h_i which guarantee stability decrease as the ratio is increased, being clearly smaller for the third order method. Moreover the stability is more sensitive to the integration step-sizes further from the reference point $tk+1$ than to h_0. So, in order to be able to manage larger values of h_i, in this work the second order variable step integration method has been used. Table I shows the intersection (L_r) of the stability plots with the negative real axis for different values of r for the second order method.

III. Fast approximation of maximum step-size for numerical stability

Given a set of linear ordinary differential equations (ODEs)

$$\dot{X}(t_k) = A_k X(t_k) \tag{6}$$

where A is negative definite and diagonally dominant, the integration method is numerically stable if the integration step size h is

$$h \leq \frac{1}{\max\limits_{r=1,\ldots,N}(\beta_{max}|a_{r,r}|)} \tag{7}$$

where $a_{r,r}$ the diagonal element in row r of A and $\beta_{max} = max(|\beta_0|,\ldots,|\beta_p|)$ the modulus of the maximum coefficient of the pth-order AdamsBashforth formula.

This technique was proposed recently [2] for fast numerical integration of state equations representing many passive systems. In such systems, the proposed fast estimate of the step-size h guarantees stability, but there is a trade off. Step sizes obtained from eq. (7) are expected to be smaller than the maximum allowed step sizes that would be obtained from the exact calculation of the Jacobians eigenvalues.

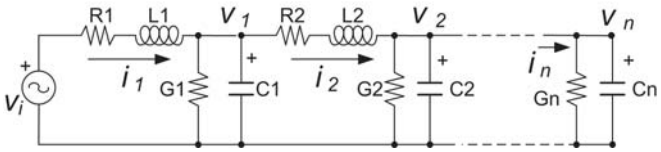

Fig. 4. Transmission line modeled as finite RLC segments.

In the case the Jacobian A is not negative definite but is symmetric, then $\max\limits_{i=1,\ldots,N}\lambda_i = ||A||$ and according to the Gershgorin theorem the values of the eigenvalues are bounded to $\lambda \leq a_{i,i} + \sum_{j=1}^{N}|a_{i,j}|$ for $i \neq j$ [2]. Given that all the Jacobian entries are real numbers, we finally obtain that [20]:

$$h \leq \frac{L_r}{\sum_{j=1}^{N} a_{i,j}} \tag{8}$$

where L_r is the intersection of the stability plot for a given step-ratio r with the negative semi-axis of the complex plane.

IV. Interconnect with inductance

Figure 4 shows an interconnect modelled as a series of finite RLC segments. Given that the currents through the inductors are state variables, the total number of state variables in an interconnect track is twice the number required for an RC interconnect model.

So, the matrix formulation of the transmission line is given by:

$$\frac{d}{dt}\begin{pmatrix} i_1 \\ v_1 \\ i_2 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = R\begin{pmatrix} i_1 \\ v_1 \\ i_2 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} \frac{1}{L_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} v_i \tag{9}$$

where the matrix R is:

$$R = \begin{pmatrix} \frac{-R_1}{L_1} & \frac{-1}{L_1} & 0 & 0 & 0 & 0 & \ldots & 0 \\ \frac{1}{C_1} & \frac{-1}{C_1 G_1} & \frac{-1}{C_2} & 0 & 0 & 0 & \ldots & 0 \\ 0 & \frac{1}{L_2} & \frac{-R_2}{L_2} & \frac{-1}{L_2} & 0 & 0 & \ldots & 0 \\ 0 & 0 & \frac{1}{C_1} & \frac{-1}{C_2 G_2} & \frac{-1}{C_2} & 0 & \ldots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & \frac{1}{C_n} & \frac{-1}{C_n G_n} \end{pmatrix} \tag{10}$$

We have performed a number of simulation tests for RLC tracks of different lengths using the proposed method and CU-SPICE, the CUDA version of SPICE [19], using the following component values per discrete section: $C = 1fF$, $L = 100pH$, $R = 10\Omega$, $G = 400\Omega^{-1}$. The excitation was a 1V step and the responses in the first RLC segment are shown in figures 5 and 6 for CU-SPICE and our method respectively.

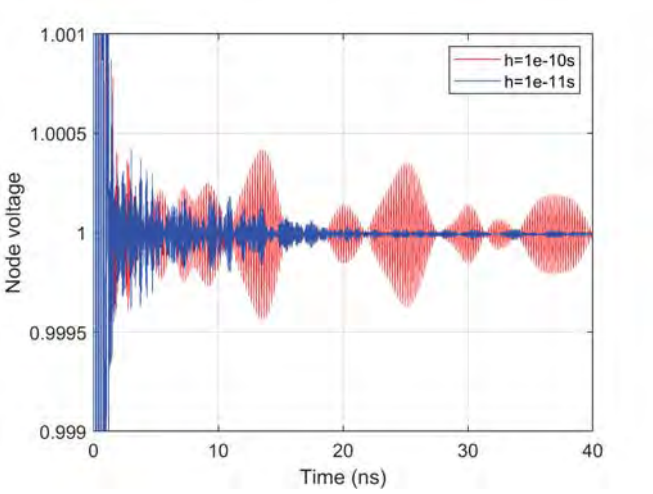

Fig. 5. SPICE simulation results for the RLC interconnect line in fig 4.

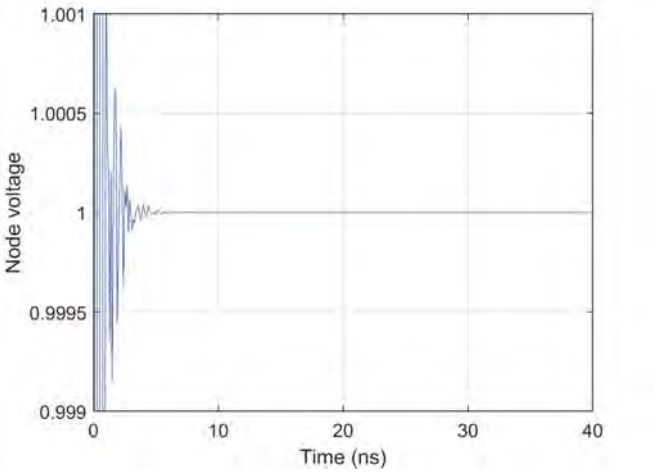

Fig. 6. Simulation results using the proposed technique for the RLC interconnect line in figure 4.

The step-size in the explicit integration was 10^{-13}sec and CU-SPICE simulations were performed using two different step-size limits: 10^{-10}sec and 10^{-11}sec. The spurious numerical ringing is evident in the CU-SPICE results and is absent, after the initial transient process, from the results obtained by the proposed method. In table II we show the CPU times for both methods, the proposed method with the step size of 10^{-13} sec and CU-SPICE with the step-size of 10^{-11}. CU-SPICE, despite using a step size larger by two orders of magnitude than that of our method, is significantly slower with the proposed method reaching a speed up of almost an order of magnitude for 10,000 RLC segments.

V. Conclusion

This paper shows promising results obtained when solving equations of large VLSI interconnect by means of explicit integration and state equations. The inclusion of analogue interconnect models in digital VLSI simulations is increasingly important as clock frequencies reach 10 GHz and more. At such speeds the analogue transients which occur in the interconnect cannot be ignored. As interconnect may constitute a large part of an analogue or a mixed-signal VLSI system, the presented method, which accelerates simulations on GPUs can be a useful approach in the development of modern VLSI design tools. For the first time, we present here results of applying the proposed method to interconnect with inductance. These results are preliminary, as this is still work in progress. The presented results are merely meant to illustrate the potential of explicit integration in the solution of vast numbers of equations representing VLSI interconnect.

TABLE II
GPU simulation time for proposed explicit method and CUspice

Segments	Explicit (s)	Implicit (s) CUspice	Speedup
100	80.294	40.537	0.504
200	85.387	48.562	0.568
500	98.019	64.159	0.654
1000	96.768	101.833	1.052
2000	100.729	196.108	1.946
5000	100.522	619.070	6.158
10000	123.041	1068.640	8.685

VI. Acknowledgements

This work has been partially funded by Spanish government through project RTI2018-097088-B-C33 (MINECO/FEDER, UE) and by EPSRC (the UK Engineering and Physical Sciences Research Council) under grant EP/N0317681/1. The research stay at University of Southampton (UK) has been supported by by Fundacin Sneca-Agencia de Ciencia y Tecnologa de la Regin de Murcia, Programa Regional de Movilidad, Colaboracin e Intercambio de Conocimiento Jimenez de la Espada under grant 21187/EE/19.

References

[1] K. C. A. Lam and M. Zwolinski. "Circuit simulation using state space equations" in Proc. 9th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME). Villach. 2013. pp. 177-180.

[2] T. J. Kazmierski. L. Wang. B. M. Al-Hashimi and G. V. Merrett. "An Explicit Linearized State-Space Technique for Accelerated Simulation of Electromagnetic Vibration Energy Harvesters." IEEE Trans. on Computer-Aided Design of Integ. Circ. and Syst. vol. 31, pp. 522-531. Apr. 2012.

[3] NVIDIA. CUDA C Programming Guide Version 7.0. Accessed: Mar. 5, 2018. [Online]. Available: http://docs.nvidia.com/cuda/cudac-programming-guide/

[4] K. Gulati, J. F. Croix, S. P. Khatri and R. Shastry, "Fast circuit simulation on graphics processing units," in Proc. Asia and South Pacific Design Automation Conference, Yokohama, 2009, pp. 403-408.

[5] R. E. Poore, "GPU-accelerated time-domain circuit simulation" in Proc. IEEE Custom Integrated Circuits Conference, Rome, 2009, pp. 629-632.

[6] L. Han and Z. Feng, "TinySPICE Plus: Scaling up statistical SPICE simulations on GPU leveraging shared-memory based sparse matrix solution techniques," in Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, 2016, pp. 1-6.

[7] E. Schneider, M. A. Kochte, S. Holst, X. Wen and H. Wunderlich, "GPU-Accelerated Simulation of Small Delay Faults," IEEE Trans. on Comp.-Aided Design of Integ. Circ. and Syst., vol. 36, no. 5, pp. 829-841, May 2017.

[8] Y. Liang, W. T. Tang, R. Zhao, M. Lu, H. P. Huynh and R. S. M. Goh, ”Scale-Free Sparse Matrix-Vector Multiplication on Many-Core Architectures,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 12, pp. 2106-2119, Dec. 2017.

[9] X. Chen, L. Ren, Y. Wang and H. Yang, ”GPU-Accelerated Sparse LU Factorization for Circuit Simulation with Performance Modeling” IEEE Trans. on Parallel and Distr. Syst., vol. 26, pp. 786-795, Mar. 2015.

[10] K. He, S. X. -. Tan, H. Wang and G. Shi, ”GPU-Accelerated Parallel Sparse LU Factorization Method for Fast Circuit Analysis,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, pp. 1140-1150, Mar. 2016.

[11] W. Lee, R. Achar and M. S. Nakhla, ”Dynamic GPU Parallel Sparse LU Factorization for Fast Circuit Simulation,” IEEE Transactions on Very Large Scale Integration Systems, vol. 26, pp. 2518-2529, Nov. 2018.

[12] O. Schenk and K. Gartner, Solving unsymmetric sparse systems of linear equations with PARDISO, Future Generat. Comput. Syst., vol. 20, no. 3, pp. 475487, Apr. 2004.

[13] T. A. Davis and E. P. Natarajan, Algorithm 907: KLU, a direct sparse solver for circuit simulation problems, ACM Trans. Math. Softw., vol. 37, no. 3, Sep. 2010, Art. no. 36.

[14] Cusolver Library, document DU-06709-001 v9.0, Jun. 2017.

[15] G. Domenech-Asensi and T. J. Kazmierski, ”An efficient numerical solution technique for VLSI interconnect equations on many-core processors”, in Proc. IEEE Intl. Symp. on Circ. and Systems, Sapporo, Japan, 2019.

[16] Ngspice Circuit Simulator. Accessed: May. 24, 2019. [Online]. Available: www.ngspice.org

[17] L. O. Chua and P. Y. Lin. Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques. Englewood Cliffs, NJ: Prentice-Hall. 1975.

[18] J. D. Hoffman, Numerical Methods for Engineers and Sciencists 2nd Ed.,CRC Press, 2001.

[19] F. Lannutti, F. Menichelli, M. Olivieri ”CUSPICE The revolutionary NGSPICE on CUDA Platforms”, in Proc. 12th MOS-AK Workshop at the ESSDERC/ESSCIRC Conference, Venice, Italy, 2014.

[20] G. Domenech-Asensi and T. J. Kazmierski, ”Stability and efficiency of explicit integration in interconnect analysis on GPUs”, accepted for publication in Proc. IEEE Intl. Symp. on Circ. and Systems, Sevilla, Spain, 2020.