An Adjoint Dynamic Neural Network Technique for Exact Sensitivities in Nonlinear Transient Modeling and High-Speed Interconnect Design

Y. Cao, J.J. Xu, V.K. Devabhaktuni, R.T. Ding*, and Q.J. Zhang

Department of Electronics, Carleton University, Ottawa, Ontario, K1S 5B6, Canada
*School of Electronics and Information Engineering, Tianjin University, Tianjin, China

Abstract — **We propose a new adjoint dynamic neural network (ADNN) technique aimed at enhancing computer-aided design (CAD) of high-speed VLSI modules. A novel formulation for exact sensitivities is derived employing Lagrange functions approach, and by defining an adjoint of a dynamic neural network (DNN), for the first time. The proposed ADNN is a dynamic model that we solve using integration backwards through time. One ADNN solution can be used to efficiently compute exact sensitivities of the corresponding DNN with respect to all its parameters. Using these sensitivities, we developed a training algorithm that facilitates DNN learning of nonlinear transients directly from continuous time-domain waveform data. Resulting accurate and fast DNN models can be straightaway used for carrying out high-speed VLSI CAD in SPICE-like time-domain environment. The technique can also speed-up physics-based nonlinear circuit CAD through faster sensitivity computations. Applications of the proposed ADNN technique in transient modeling and nonlinear design are demonstrated through high-speed interconnect driver examples.**

I. INTRODUCTION

Over the last decade, a CAD approach based on artificial neural networks (ANN) gained recognition in RF, microwave and VLSI community [1]. Fast neural models trained using appropriate input-output data can accurately represent device or circuit behaviors [2]. A most recent trend in this area is the use of a unique category of ANN called DNN [3] for dynamic nonlinear CAD. In [3], DNN have been shown to address large-signal modeling and design in the case of nonlinear periodic RF/microwave responses in harmonic balance (HB) environment [4].

In this paper, we further expand DNN into an even more powerful technique that can handle nonlinear transients. Accurate and fast representation of nonlinear transient behaviors is a key to successful digital high-speed VLSI interconnect CAD including multi-chip modules and multi-layer PCBs. Extensive research has been conducted on modeling and simulation of interconnect networks resulting in both circuit- and ANN-based techniques [5][6]. Currently, CAD of interconnect modules with nonlinear terminations is an active research subject [7] and this paper is targeted toward accomplishing efficient neural based nonlinear transient modeling and design of high-speed interconnect components including physics-based effects.

In order for the DNN to learn transient data, sensitivities (derivatives) of the corresponding training error w.r.t. DNN weights are essential. On the other hand, transient design requires sensitivities of the target functions w.r.t. geometrical/physical parameters of nonlinear components. A paramount theory for sensitivity analysis is the adjoint sensitivity technique pioneered in [8]. Following this, various adjoint sensitivity techniques such as HB-based [4], circuit-based [9], and static ANN-based [10] techniques have been developed. In this paper, exact adjoint sensitivity for DNN is investigated to enable transient modeling and design.

For the first time, we propose a new ADNN technique for nonlinear transients such as those in high-speed interconnects with nonlinear terminations. An elegant formulation for exact adjoint sensitivities is derived employing Lagrange functions approach, by defining an adjoint of the DNN. These formulae are exploited to engineer a robust training algorithm that facilitates efficient DNN training directly from transient data. These sensitivities can also lead to faster nonlinear circuit design including geometrical/physical parameters. The proposed technique is demonstrated through DNN- modeling and circuit design examples of nonlinear interconnect drivers.

II. DYNAMIC NEURAL NETWORKS

In time-domain, a nonlinear circuit can be represented by a DNN [3] of order n. Inputs to DNN include dynamic inputs $\boldsymbol{u}(t)$, corresponding k^{th}-order derivatives $\boldsymbol{u}^{(k)}(t)$, and static inputs $\boldsymbol{p}$. Here, $\boldsymbol{p}$ is a n_p-vector including geometrical/physical parameters (e.g., channel width). The DNN equations of the nonlinear circuit are given by,

$$\begin{aligned} \dot{\boldsymbol{v}}_1(t) &= \boldsymbol{v}_2(t) \\ &\vdots \\ \dot{\boldsymbol{v}}_{n-1}(t) &= \boldsymbol{v}_n(t) \\ \dot{\boldsymbol{v}}_n(t) &= \boldsymbol{f}_{\text{ANN}}(\boldsymbol{p}, \boldsymbol{u}^{(n-1)}(t),\ldots,\boldsymbol{u}^{(1)}(t),\boldsymbol{u}(t),\boldsymbol{v}_n(t),\ldots,\dot{\boldsymbol{v}}_1(t),\boldsymbol{w}) \end{aligned} \quad ,(1)$$

where each $\boldsymbol{v}_i$ is a n_y-vector representing a state of the DNN. Here, $\boldsymbol{f}_{\text{ANN}}$ represents a MLP neural network [1][2] with trainable weight parameters $\boldsymbol{w}$. Output signals of the DNN model are given by $\boldsymbol{y}(t) = \boldsymbol{v}_1(t)$.

0-7803-7695-1/03/$17.00 © 2003 IEEE

III. Proposed ADNN Sensitivity Technique

We define an energy function E for DNN that represents a typical design function in transient analysis [9] as

$$E = \int_{T_1}^{T_2} f(\boldsymbol{y}(t))\ dt\,, \tag{2}$$

where $[T_1, T_2]$ is the time-interval of interest. The purpose of our sensitivity technique is to determine the derivatives of E w.r.t. DNN weights $\boldsymbol{w}$ and static inputs $\boldsymbol{p}$. The challenge here is that, $\boldsymbol{y}(t)$ has a dynamic and not a simple algebraic relationship with $\boldsymbol{w}$ and $\boldsymbol{p}$. For sensitivity derivation, we define a Lagrange function L as

$$L = f(\boldsymbol{y}(t)) + \hat{\boldsymbol{v}}_n^{\,T}(\dot{\boldsymbol{v}}_n - \boldsymbol{f}_{\text{ANN}}) + \sum_{j=1}^{n-1} \hat{\boldsymbol{v}}_j^{\,T}(\dot{\boldsymbol{v}}_j - \boldsymbol{v}_{j+1})\,, \tag{3}$$

where each $\hat{\boldsymbol{v}}_j$ represents time-dependent Lagrange parameters independent of $\boldsymbol{w}$ and $\boldsymbol{p}$. Subject to (1), derivatives of E can be expressed in terms of L in (3) as

$$\begin{aligned}\frac{dE}{d\boldsymbol{w}} &= \hat{\boldsymbol{v}}(t)^T \frac{d\boldsymbol{v}(t)}{d\boldsymbol{w}}\bigg|_{T_2}^{T_1} - \int_{T_1}^{T_2} \hat{\boldsymbol{v}}_n^{\,T} \frac{\partial \boldsymbol{f}_{\text{ANN}}}{\partial \boldsymbol{w}}\ dt \\ &+ \int_{T_1}^{T_2} \left(\frac{\partial f}{\partial \boldsymbol{y}^T} - \dot{\hat{\boldsymbol{v}}}_1^{\,T} - \hat{\boldsymbol{v}}_n^{\,T} \frac{\partial \boldsymbol{f}_{\text{ANN}}}{\partial \boldsymbol{v}_1^{\,T}} \right) \frac{d\boldsymbol{v}_1}{d\boldsymbol{w}}\ dt \\ &+ \int_{T_1}^{T_2} \sum_{j=2}^{n} \left(-\dot{\hat{\boldsymbol{v}}}_j^{\,T} - \hat{\boldsymbol{v}}_n^{\,T} \frac{\partial \boldsymbol{f}_{\text{ANN}}}{\partial \boldsymbol{v}_j^{\,T}} - \hat{\boldsymbol{v}}_{j-1}^{\,T} \right) \frac{d\boldsymbol{v}_j}{d\boldsymbol{w}}\ dt \end{aligned} \tag{4}$$

As (4) holds good for arbitrary choice of $\hat{\boldsymbol{v}}_j$'s, we propose a new adjoint-DNN or ADNN as

$$\begin{aligned} \dot{\hat{\boldsymbol{v}}}_1 &= -\frac{\partial \boldsymbol{f}_{\text{ANN}}^{\,T}}{\partial \boldsymbol{v}_1} \hat{\boldsymbol{v}}_n + \frac{\partial f}{\partial \boldsymbol{y}} \\ \dot{\hat{\boldsymbol{v}}}_2 &= -\frac{\partial \boldsymbol{f}_{\text{ANN}}^{\,T}}{\partial \boldsymbol{v}_2} \hat{\boldsymbol{v}}_n - \hat{\boldsymbol{v}}_1 \\ &\vdots \\ \dot{\hat{\boldsymbol{v}}}_n &= -\frac{\partial \boldsymbol{f}_{\text{ANN}}^{\,T}}{\partial \boldsymbol{v}_3} \hat{\boldsymbol{v}}_n - \hat{\boldsymbol{v}}_{n-1} \end{aligned} \tag{5}$$

where each $\hat{\boldsymbol{v}}_j$ is a n_y-vector representing a state of the ADNN. As can be seen from (1) and (5), both DNN and corresponding ADNN have the same number of states. Input to ADNN (i.e. $\frac{\partial f}{\partial \boldsymbol{y}}$) excites state $\hat{\boldsymbol{v}}_1$ that corresponds to output of the original-DNN. Outputs of the ADNN are given by $\hat{\boldsymbol{y}}(t) = \hat{\boldsymbol{v}}_n(t)$. Using (5) together with a boundary condition $\hat{\boldsymbol{v}}(T_2) = \boldsymbol{0}$, equation (4) can be expressed in terms of ADNN output $\hat{\boldsymbol{y}}(t)$ and MLP network $\boldsymbol{f}_{\text{ANN}}$ in original DNN as

$$\frac{dE}{d\boldsymbol{w}} = -\int_{T_1}^{T_2} \hat{\boldsymbol{y}}^T \frac{\partial \boldsymbol{f}_{\text{ANN}}}{\partial \boldsymbol{w}}\ dt \tag{6}$$

where $\frac{\partial \boldsymbol{f}_{\text{ANN}}}{\partial \boldsymbol{w}}$ is a Jacobian matrix containing derivatives of the MLP w.r.t. individual weight parameters in $\boldsymbol{w}$.

For example, consider $\boldsymbol{f}_{\text{ANN}}$ being a 3-layer MLP. Let N_l denote number of neurons in l^{th} layer. Let w_{ij}^l represent weight of the link between j^{th} neuron of l-1^{th} layer and i^{th} neuron of l^{th} layer. We define $z_i^l(t)$ as instantaneous output of the i^{th} neuron in the l^{th} layer. Sensitivities in (6) can then be systematically evaluated as

$$\frac{dE}{dw_{ij}^l} = \begin{cases} -\int_{T_1}^{T_2} \hat{y}_k(t)\ z_i^{l-1}(t)\,dt, l = 3, 1 \le k = i \le n_y \\ -\sum_{k=1}^{n_2} w_{ki}^{l+1} \int_{T_1}^{T_2} \hat{y}_k(t)\ z_i^l(t)(1 - z_i^l(t))\, z_j^{l-1}(t)\,dt, l = 2 \end{cases}, \tag{7}$$

where $\hat{y}_k(t)$ represents k^{th} ADNN output. Sensitivity of E w.r.t. i^{th} static DNN input p_i can be evaluated using

$$\frac{dE}{dp_i} = -\sum_{k=1}^{n_y} \sum_{j=1}^{N_2} w_{kj}^3 w_{ji}^2 \int_{T_1}^{T_2} \hat{y}_k(t)\, z_j^2(t)(1 - z_j^2(t))\,dt\,. \tag{8}$$

IV. ADNN Sensitivities for DNN Training

Our elegant choice of E allows direct utilization of ADNN sensitivity formulation in section III, for DNN training using nonlinear transients. Let $\boldsymbol{u}_d^i(t)$ and $\boldsymbol{y}_d^i(t)$ represent i^{th} input and output waveforms sufficiently sampled in the time-interval $[T_1, T_2]$, to be used as training data. The objective of DNN training is to adjust DNN parameters $\boldsymbol{w}$ such that the training error function

$$E_d = \int_{T_1}^{T_2} \sum_{i=1}^{N_T} \frac{1}{2} \left\| \boldsymbol{y}^i(t) - \boldsymbol{y}_d^i(t) \right\|^2 dt \tag{9}$$

is minimized. Here, $\boldsymbol{y}^i(t)$ represents DNN prediction of i^{th} output signal and N_T represents total number of training waveforms pairs $(\boldsymbol{u}_d^i(t), \boldsymbol{y}_d^i(t))$. Since f in (2) can be any function, we judiciously choose $f(\boldsymbol{y}(t))$ as

$$f(\boldsymbol{y}(t)) = \sum_{i=1}^{N_T} \frac{1}{2} \left\| \boldsymbol{y}^i(t) - \boldsymbol{y}_d^i(t) \right\|^2 \tag{10}$$

to establish consistency between energy function E and training error E_d. Dynamic training process for i^{th} training waveform pair $(\boldsymbol{u}_d^i(t), \boldsymbol{y}_d^i(t))$ is explained here. Given $\boldsymbol{u}_d^i(t)$, original DNN equations in (1) are integrated forward from time T_1 up to T_2 with user-specified initial condition $\boldsymbol{v}(T_1)$, resulting in current DNN outputs $\boldsymbol{y}^i(t)$. We substitute $\boldsymbol{y}^i(t)$ in (10) and then integrate ADNN equations in (5) backwards from time T_2 down to T_1, setting $\hat{\boldsymbol{v}}(T_2) = \boldsymbol{0}$. Finally, dynamic error sensitivities computed using (7) are supplied to a gradient-based

training algorithm (e.g. quasi-Newton) for determining the weight update during DNN training process shown in Fig. 1.

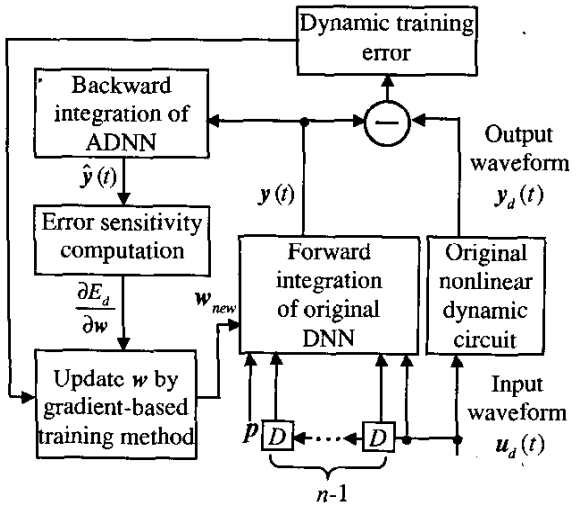

Fig. 1. Block diagram of the DNN training exploiting exact sensitivities from proposed ADNN. D is a symbol for differentiation.

V. Examples

A. CMOS Inverter

This example is for validating the proposed ADNN sensitivity technique. A DNN is trained to learn transient behaviors of the CMOS inverter and error derivatives needed during training are computed via our ADNN. Inputs to the DNN model include dynamic input v_{in} (input voltage) and static input W (transistor size). The model output is v_{out} (output voltage). Training waveforms are generated using level-49 BSIM3V3 *HSPICE* model. A DNN of order $n = 1$ including a 3-layer MLP with 8 hidden neurons is used. Sensitivities of the dynamic training error function w.r.t. DNN parameters using the proposed ADNN technique accurately match those from perturbation method as shown in Table I, thus validating the proposed ADNN.

Table I. Comparison of Sensitivity Values.

DNN Sensitivity	Perturbation Method	Proposed ADNN	Difference (as %)
$\partial E_d / \partial w_{11}^3$	-2.4327e-02	-2.4336e-02	0.036
$\partial E_d / \partial w_{13}^3$	+1.5897e-03	+1.5904e-03	0.044
$\partial E_d / \partial w_{15}^3$	+6.2217e-04	+6.2241e-04	0.038
$\partial E_d / \partial w_{17}^3$	-1.5563e-05	-1.5545e-05	0.116
$\partial E_d / \partial W$	-1.5638e+01	-1.5748e+01	0.703

B. Circuit-Based High-Speed Interconnect Driver

This practical example shows the utility of the proposed ADNN sensitivity technique in transient modeling. A DNN model of a CMOS inverter driver is developed. Dynamic and static DNN inputs are $\boldsymbol{u} = [v_{in}, i_{out}]^T$ and $\boldsymbol{p} = [W_N]^T$ respectively, and DNN output is $\boldsymbol{y} = [v_{out}]^T$. Two sets of training waveforms, namely, signal data and crosstalk data are generated replacing each transistor in the driver circuit by level-49 BSIM3V3 *HSPICE* model. Signal data is collected by supplying the driver with different values for pulse rise-time [0.1 0.5ns] and pulse amplitude [2.3 2.7V], and varying interconnect-length d [2 5cm] and CMOS channel width W_N [100 240μm]. Crosstalk data is generated for different pulse amplitudes [-1.0 3.0V] using the above rise-time and W_N ranges. Time-interval of interest is [0 6ns]. DC values of input signals are used as initial conditions for integrating the DNN forward.

DNN's of different dynamic orders (n) and with different f_{ANN} are trained using exact error derivatives of the proposed ADNN, and the model accuracies are shown in Table II. A single ADNN evaluation is sufficient for computing all the derivatives for each training waveform pair. Trained DNN model of order 2 with 50 hidden neurons is then used 3 times in the interconnect circuit of Fig. 2. As can be seen in Fig. 3, excellent agreement is achieved between *HSPICE* simulations and our DNN-based interconnect simulations.

Table II: DNN model Accuracies For Different Cases

No. of hidden neurons for DNN of order 2	Avg. test error
30	1.05%
50	0.42%
70	0.71%

Order of DNN with 50 hidden neurons	Avg. Test error
1	0.38%
2	0.29%
3	0.85%

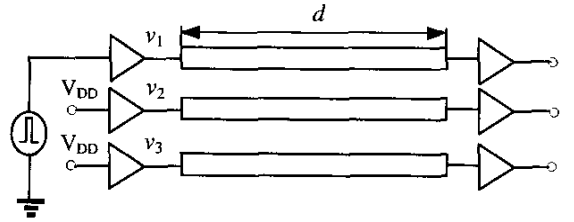

Fig. 2. A 3-conductor interconnect circuit loaded with nonlinear buffers used for generating test waveforms.

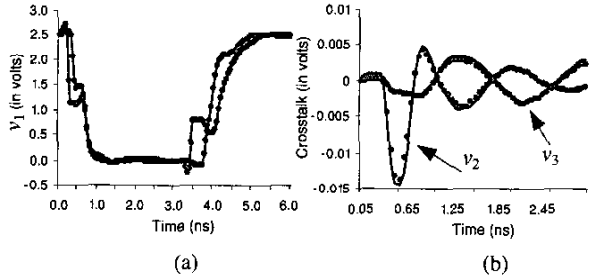

Fig. 3. Comparison of interconnect simulation of Fig. 2 using DNN models (o) and *HSPICE* (—). (a) Output signal v_1 under two different excitations and (b) Crosstalk signals v_2 and v_3.

C. Physics-Based Multistage Driver

In this example, we demonstrate the relevance of the proposed ADNN technique for physics-based transient design purpose. A 1μm 4-stage CMOS driver in Fig. 4 is considered. Training data is obtained using physics-based *MINIMOS* simulator [11]. The driver load is a single transmissionline with parameters R=36Ω/m, L=360nH/m, C=100pF/m and G=0.01S/m, and terminated with a 5pF capacitor. The DNN has the same inputs and outputs as those in example *B*. Training waveforms are generated for different values of rise-time T_r [0.25 0.75ns] and pulse amplitude A [4.5 5.5V], and varying interconnect-length d [0.08m 0.14m]. Driver size is also perturbed 50% around the nominal value. A DNN structure of dynamic order 1 and 30 hidden neurons in f_{ANN}, trained using the derivatives from the proposed ADNN technique resulted in a DNN model with average test error of 0.25%. DNN model's prediction of output voltages with independent test waveforms match very well as shown in Fig. 5, validating our transient DNN modeling.

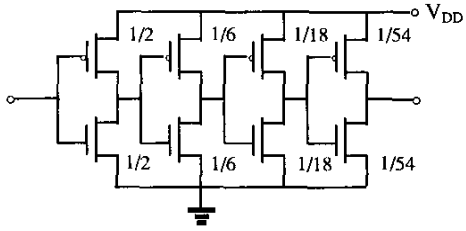

Fig. 4. A 4-stage CMOS driver to be modeled by the proposed ADNN technique using physics-based data from *MINIMOS*.

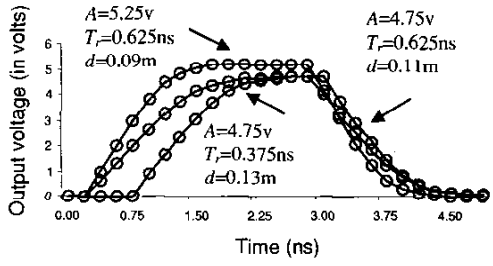

Fig. 5. Comparison of multi-stage CMOS driver voltage outputs from DNN (o) trained using proposed ADNN and test data (—).

Electrical power is an important criterion for high-speed digital design. We use our ADNN technique for computing sensitivity of average output power of the driver w.r.t. driver-size under transient excitation. Sensitivities obtained using the proposed ADNN accurately match those from physics-based *MINIMOS* perturbations as shown in Fig. 6. Total CPU-time taken by the ADNN is 2*s* as compared to 6254*s* taken by the *MINIMOS*, proving the significance of our proposed ADNN technique in nonlinear transient design.

CONCLUSIONS

A new ADNN technique for exact adjoint sensitivities of DNN's in transient environment has been proposed. The technique has been used to efficiently train DNN's for learning transient behaviors of nonlinear high-speed interconnect circuits, for the first time. The technique facilitates a means of providing sensitivities of electrical criteria w.r.t. geometrical/physical design parameters, with physics-level accuracies but only requiring tiny fraction of the CPU-time taken by physics-based sensitivity computations. This work is significant for efficient modeling simulation and design of high-speed VLSI interconnect modules under transient excitations.

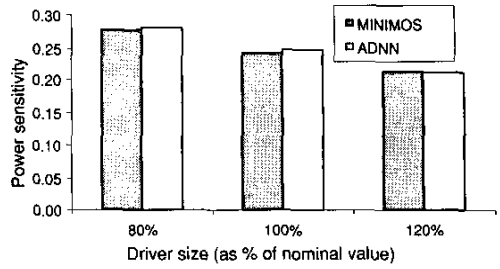

Fig. 6. Comparison of power sensitivities of multi-stage driver computed using proposed ADNN and physics-based *MINIMOS*.

REFERENCES

[1] Q.J. Zhang and K.C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA: Artech House, 2000.

[2] Q.J. Zhang, K.C. Gupta, and V.K. Devabhaktuni, "Artificial neural networks for RF and microwave design: From theory to practice," *IEEE Trans. Microwave Theory Tech.*, vol. 51, 2003. (Accepted).

[3] J.J. Xu, M. Yagoub, R.T. Ding, and Q.J. Zhang, "Neural based dynamic modeling of nonlinear microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 2769-2780, 2002.

[4] J.W. Bandler, Q.J. Zhang, and R.M. Biernacki, "A unified theory for frequency-domain simulation and sensitivity analysis of linear and nonlinear circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 1661-1669, 1988.

[5] R. Achar and M.S. Nakhla "Simulation of high-speed interconnects," *Proc. IEEE*, vol. 89, pp. 693-728, 2001.

[6] A. Veluswami, M.S. Nakhla, and Q.J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 712-723, 1997.

[7] A. Dounavis, R. Achar, and M.S. Nakhla "Efficient sensitivity analysis of lossy multiconductor transmission lines with nonlinear terminations," *IEEE Trans. Microwave Theory Tech.*, vol. 49, pp. 2292-2299, 2001.

[8] S.W. Director and R.A. Rohrer, "The generalized adjoint network and network sensitivies," *IEEE Trans. Circuit Theory*, vol. 16, pp. 318-323, 1969.

[9] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York, NY: VNR, 1993.

[10] J.J. Xu, M. Yagoub, R.T. Ding, and Q.J. Zhang, "Exact adjoint sensitivity analysis for neural based microwave modeling and design," *IEEE Trans. Microwave Theory Tech.* vol. 51, 2003. (Accepted).

[11] *MINIMOS-NT v.2.0*, Institute for Microelectronics, Technical University Vienna, Vienna, Austria.