IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 25, 2021, accepted November 26, 2021, date of publication November 30, 2021,
date of current version December 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3131730

Thin-Film Transistor Simulations With the Voltage-In-Current Latency Insertion Method

WEI CHUN CHIN[1], ANDREI PASHKOVICH[2], (Member, IEEE), JOSÉ E. SCHUTT-AINÉ[3], (Fellow, IEEE), NUR SYAZREEN AHMAD[1], (Member, IEEE), AND PATRICK GOH[1]

[1]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, Penang 14300, Malaysia
[2]Silvaco, Inc., Santa Clara, CA 95054, USA
[3]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Corresponding author: Patrick Goh (eepatrick@usm.my)

This work was supported in part by the Ministry of Higher Education, Malaysia, through the Fundamental Research Grant Scheme (FRGS) under Grant FRGS/1/2020/TK0/USM/02/7; in part by the U.S. Army Small Business Innovation Research (SBIR) Program Office; in part by the U.S. Army Research Office under Contract W911NF-16-C-0125, and in part by Silvaco, Inc.

ABSTRACT This article presents formulations for the voltage-in-current (VinC) latency insertion method (LIM) for thin-film transistors (TFTs). LIM is a fast circuit simulation algorithm that solves circuits in a leapfrog manner, without requiring intensive matrix operations present in SPICE-based simulators. This allows LIM to have a far superior scaling with respect to the size of the circuit resulting in significant time savings on large circuit networks. The VinC LIM formulation for the TFTs written in this article has the benefit of a better stability compared to the original LIM formulation which allows the use of larger time steps. The performance of the new algorithm is demonstrated through the simulation of numerical examples of large flat-panel display (FPD) circuits. It is seen that VinC LIM greatly outperforms basic LIM and commercial SPICE-based simulators, where the presented algorithm is able to simulate circuits with more than 10 million nodes or devices in a reasonable time, which is not viable in many modern day SPICE-based simulators.

INDEX TERMS Circuit analysis, latency insertion method, thin-film transistor.

I. INTRODUCTION

As the drive for higher functionality, capacity, and efficiency continues to push the consumer industry in IC products forward, the complexity and sheer size of the analog and digital circuitry inside these devices continue to rise in tandem as well. In order for a design to succeed, a mature product with the latest process technology node requires support during every step of the development process. From front-end design steps, to back-end packaging and test procedures, all sections of the design and verification process have to be equipped with the latest technology with sufficient capabilities to avoid potential failures. This presents a big challenge in the semiconductor industry today as product designs need to be signed-off, by passing through various modeling and simulation verifications, before being sent to the foundry to reduce the time and manufacturing cycles with the ultimate goal of minimizing development costs.

In this regard, access to accurate and fast modeling and simulation tools is paramount to the success of a product. Most of the commercial tools on circuit simulations today rely on the Simulation Program with Integrated Circuit Emphasis (SPICE) [1] algorithm, which was originally developed in 1973, based on the modified nodal analysis (MNA) framework, with a heavy reliance of sparse matrix solvers. However, with the continued scaling of process technology nodes, more and more devices are now crammed into a single product, where a full circuit design can contain tens or even hundreds of millions of device elements. This creates a simulation bottleneck, where state-of-the-art simulators are no longer able to cope with the sheer large size of the resulting MNA matrices from these circuits, thus opening the door to new methods which could potentially replace the presently aging simulation algorithms.

The latency insertion method (LIM) was originally developed as an alternative to SPICE-based solvers for the fast transient simulation of large networks [2]. Since then, various improvements to the algorithm have been presented, which includes the block-LIM [3], partitioned

The associate editor coordinating the review of this manuscript and approving it for publication was Dušan Grujić.

LIM (PLIM) [4], predictor-corrector LIM [5], locally implicit LIM (LILIM) [6], alternating direction explicit-LIM (ADE-LIM) [7], and the voltage-in-current LIM (VinC LIM) [8]. These algorithms have been applied in a myriad of applications, including power supply networks [9]–[12], transmission lines [13], [14], electro-thermal analysis [15], I/O buffers [16], [17], phase-locked loops [18], and power electronic systems [19], [20].

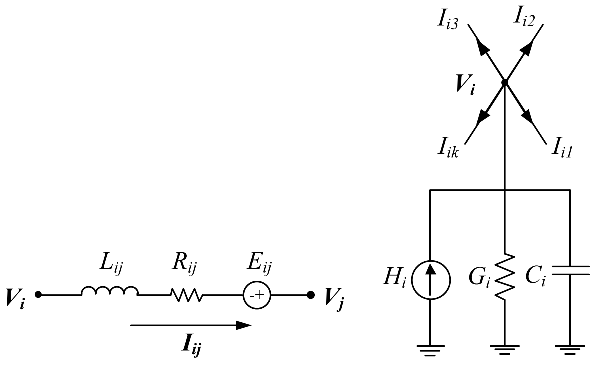
Recently, thin-film transistors (TFTs) have grown in prominence as the standard device in display technology particularly in flat panel displays (FPDs) based on the organic light-emitting diode (OLED) display technology [21]–[23]. In these active-matrix devices, individual active switching elements are used to control each of the pixel in the panel. As a result, the simulation of these circuits are incredibly resource consuming from a runtime perspective as they contain millions of transistors, which make up the array of pixels for a given resolution. For example, a full HD (1080p) display can contain more than 40 million TFTs, and this number goes up even higher in 4K and 8K displays. These massive circuits are far beyond the capacity of state-of-the-art commercial simulators, which often struggle in both runtime and memory requirements due to the poor scaling of their matrix based operations.

In this article, formulations for the simulations of TFTs in the LIM framework are presented. First, the background and fundamentals of the original LIM and VinC LIM are reviewed. Then, augmentations to the formulations are presented for TFTs, which include the drain-source current, and the gate-drain and gate-source intrinsic capacitance currents. Practical steps are discussed to control the accuracy and improve the runtime. Finally, numerical results are presented, including large FPD arrays of varying resolutions, to illustrate the viability of the method for the simulations of very large circuits. Results show that the proposed TFT VinC LIM simulation algorithm is able to produce accurate simulation waveforms, while having a capacity far exceeding the original LIM simulation algorithm and state-of-the-art SPICE-based commercial simulators.

The rest of the paper is organized as follows. In Section II, the fundamentals of the original LIM and VinC LIM algorithms are reviewed. Then, in Section III, new formulations of the LIM algorithms for TFTs are derived in both the original LIM and VinC LIM frameworks. In Section IV, numerical results are presented for different TFT circuits, and the performance of the developed VinC LIM TFT algorithm is compared with the original LIM algorithm and a current generation SPICE simulator in terms of accuracy and speed. Finally, Section V concludes the article and proffers some future work on the subject.

II. LIM FORMULATION

The fundamentals of basic LIM and VinC LIM are reviewed in this section.



(a) (b)

FIGURE 1. Circuit topology required for LIM simulation. (a) Branch topology. (b) Node topology.

A. BASIC LIM FORMULATION

The original LIM or basic LIM is a circuit simulation algorithm that is devised for the solution of large and dense networks. To analyze a circuit with LIM, the circuit must be composed of nodes and branches with topologies as shown in Fig. 1, where V_i and V_j are node voltages at node i and node j respectively, I_{ij} is the branch current flowing from node i to j, L_{ij}, R_{ij}, and E_{ij} are the branch inductance, resistance, and voltage source respectively, C_i, G_i, and H_i are the node capacitance, conductance, and current source respectively, and the currents labeled from I_{i1} to I_{ik} are the branch currents that are connected to node i.

From the circuit topology in Fig. 1, the LIM algorithm is formed by applying Kirchhoff's laws, where the inductance and capacitance in each topology results in a derivative term in their equations, which are linearized through the application of Euler's method. By solving the equations, the basic LIM formulations can be obtained as

$$I_{ij}^{n+1} = I_{ij}^{n} + \frac{\Delta t}{L_{ij}}\left(V_i^{n+\frac{1}{2}} - V_j^{n+\frac{1}{2}} - R_{ij}I_{ij}^{n} + E_{ij}^{n+\frac{1}{2}}\right) \tag{1}$$

$$V_i^{n+\frac{1}{2}} = V_i^{n-\frac{1}{2}} + \frac{\Delta t}{C_i}\left(H_i^{n} - G_iV_i^{n-\frac{1}{2}} - \sum_{k=1}^{M_i} I_{ik}^{n}\right) \tag{2}$$

where M_i indicates the total number of branches connected to node i, Δt is the time step of the simulation, and n is the time index which marks the time point in the transient analysis. In a LIM simulation, the leapfrog concept is applied where (1) and (2) are performed in an alternating manner throughout the transient flow. The leapfrog updating pattern is indicated by the half time step difference in the indices of the voltage and current equations where the voltage values are solved for a half time step before the current values are solved for. Small values of latencies (capacitances and inductances) are required in order to perform this time stepping solution, and fictitious values are inserted into the circuit if they are not present.

Compared to SPICE based methods, which make use of MNA stamps and implicit simultaneous solutions of equations, the LIM simulation is performed in an explicit manner.

This gets rid of heavy matrix computations and hence, allows the LIM analysis to scale better with the size of the circuit and be carried at a relatively faster speed, especially for very large networks. However, since Euler's approximation is used, the selection of the time step becomes a limiting factor for the stability of the simulation. Through the application of Lyapunov's method [24], the maximum allowable time step value for stability can be calculated by

$$\Delta t_{\max} = \sqrt{2} \min_{i=1}^{n_N} \left(\sqrt{\frac{C_i}{n_B^i} \min_{m=1}^{n_B^i} L_{im}} \right) \tag{3}$$

where n_N is the number of nodes in the circuit, n_B^i is the number of branches connected to node i, C_i is the value of the shunt capacitor at node i, and L_{im} is the value of the inductor in the m^{th} branch connected to node i. Based on (3), it can be observed that the maximum stable time step value of a LIM simulation depends on the minimum value of the latencies in the circuit. Therefore, there is always a trade-off between simulation time and accuracy in a LIM simulation if fictitious components are required to be inserted into the circuit. Bigger fictitious components lead to a drop in accuracy while smaller values lead to a significant increase in total simulation time as the maximum stable time step is reduced to very small values.

B. VOLTAGE-IN-CURRENT LIM FORMULATION

The main limitation of LIM is its conditional stability which results in the use of very small time steps. To solve this, various improved versions of LIM have been introduced. VinC LIM [8] is an improved formulation of LIM for better stability and it is formulated by rewriting equations (1) and (2) in their respective implicit versions at the same time step of $n+1$. These equations are given by

$$V_i^{n+1} = \frac{\frac{C_i}{\Delta t} V_i^n - \sum_{k=1}^{M_i} I_{ik}^{n+1} + H_i^{n+1}}{\frac{C_i}{\Delta t} + G_i} \tag{4}$$

$$I_{ij}^{n+1} = \frac{\frac{L_{ij}}{\Delta t} I_{ij}^n + V_i^{n+1} - V_j^{n+1} + E_{ij}^{n+1}}{\frac{L_{ij}}{\Delta t} + R_{ij}}. \tag{5}$$

Then, VinC LIM is formed by substituting the voltage equation in (4) into (5) and solving the substituted equation for the I_{ij}^{n+1} terms. The resulting current equation in VinC LIM is given by

$$I_{ij}^{n+1} = \frac{\frac{L_{ij}}{\Delta t} I_{ij}^n + \frac{\frac{C_i}{\Delta t} V_i^n + H_i^{n+1} - \sum_{k=1,k\neq j}^{M_i} I_{ik}^{n+p}}{\frac{C_i}{\Delta t} + G_i} - \frac{\frac{C_j}{\Delta t} V_j^n + H_j^{n+1} - \sum_{k=1,k\neq i}^{M_j} I_{jk}^{n+p}}{\frac{C_j}{\Delta t} + G_j} + E_{ij}^{n+1}}{R_{ij} + \frac{L_{ij}}{\Delta t} + \left(G_i + \frac{C_i}{\Delta t}\right)^{-1} + \left(G_j + \frac{C_j}{\Delta t}\right)^{-1}} \tag{6}$$

where $\sum_{k=1,k\neq j}^{M_i} I_{ik}^{n+p}$ is the sum of currents flowing out of node i, and $\sum_{k=1,k\neq i}^{M_j} I_{jk}^{n+p}$ is the sum of currents flowing out of node j, without considering the I_{ij} current in both summations. In order to circumvent the added complexity of the implicit formulation, a forward branch-marching scheme is applied in VinC LIM, where the sequence of the currents in the updating process is taken into consideration, and the latest available current value is always used. This is indicated by the indices p that are shown in the equation, where p is either 1 or 0, depending on whether the associated current has been solved for or not, before the present computation at this time step. Compared to the explicit derivation of the basic LIM formulation, the implicit derivation of the VinC LIM formulation leads to improvements in the stability of the algorithm. Previous results on VinC LIM have shown the method to be unconditionally stable, thus allowing the use of timesteps far greater than that which is permitted in basic LIM.

III. LIM FORMULATION FOR RPI THIN-FILM TRANSISTORS

Both the basic LIM and VinC LIM formulations discussed in the previous section are limited to the simulation of linear circuits only. Hence, more work is required to extend the coverage of the LIM formulations to other devices especially transistors in order to support the simulations of various products. In this work, the formulations for basic LIM and VinC LIM for thin-film transistors (TFTs) are presented. TFTs are the main active devices which are commonly used in display panels such as those based on the organic light-emitting diode (OLED) display technology. This section covers the formulation for the drain-source current and the gate-drain and gate-source intrinsic capacitance currents based on the Rensselaer Polytechnic Institute (RPI) TFT model using LIM's formulation.

A. BASIC LIM FORMULATION FOR RPI TFT DRAIN-SOURCE CURRENT

For circuit simulations, the operation of a TFT device is mainly classified into two parts, which are the calculation of the drain-source current, I_{ds}, also known as the main current, and the intrinsic capacitance currents made up by the gate-drain current, I_{gd} and the gate-source current, I_{gs}. Fig. 2 shows a simplified equivalent circuit of a RPI TFT model where V_d, V_g, and V_s represent the node voltages at the drain, gate, and source nodes of the TFT respectively, the three arrows represent the default directions of I_{ds}, I_{gd}, and I_{gs} current flows, and C_{gd} and C_{gs} are the intrinsic capacitances that exist between the gate-drain and gate-source terminals respectively. In this part, the formulation of the I_{ds} current in the basic LIM formulation is discussed.

In basic LIM, the voltage equations at the three different terminals of the RPI TFT model are given by

$$V_d^{n+\frac{1}{2}} = \frac{\frac{C_d}{\Delta t} V_d^{n-\frac{1}{2}} - \sum_{k=1}^{M_d} I_{dk}^n + H_d^n}{\frac{C_d}{\Delta t} + G_d} \tag{7}$$

$$V_s^{n+\frac{1}{2}} = \frac{\frac{C_s}{\Delta t} V_s^{n-\frac{1}{2}} - \sum_{k=1}^{M_s} I_{sk}^n + H_s^n}{\frac{C_s}{\Delta t} + G_s} \tag{8}$$

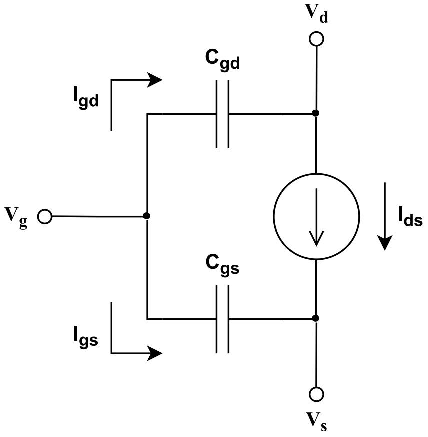


FIGURE 2. Equivalent circuit of a RPI thin-film transistor.

$$V_g^{n+\frac{1}{2}} = \frac{\frac{C_g}{\Delta t} V_g^{n-\frac{1}{2}} - \sum_{k=1}^{M_g} I_{gk}^n + H_g^n}{\frac{C_g}{\Delta t} + G_g} \tag{9}$$

where V_d, V_s, and V_g represent the voltages at the drain, source, and gate nodes of the TFT at their respective indices of $n + \frac{1}{2}$ and $n - \frac{1}{2}$. C is the shunt capacitance, G is the conductance, and H is the current source at the nodes labeled according to their respective nodes d, s, and g, and M_d, M_s, and M_g represent the number of branches connected to nodes d, s, and g respectively.

In the basic LIM formulation, the drain-source current formula of the TFT model can be applied directly in the equations. Taking an example using version 1.0 of the RPI poly-Si TFT model, the drain-source current equation is given by

$$I_{ds} = \frac{I_a \cdot I_{sub}}{I_a + I_{sub}} (1 + \Delta_{kink}) + I_{leak} \tag{10}$$

where I_{sub} is the subthreshold current, I_a is the above threshold current, Δ_{kink} takes into account the kink effect, and I_{leak} is the leakage current. By adding suitable indices to the current terms in the equation, the basic LIM current equation for the version 1.0 RPI poly-Si TFT drain-source current is given by

$$I_{ds}^{n+1} = \frac{I_a^{n+1} \cdot I_{sub}^{n+1}}{I_a^{n+1} + I_{sub}^{n+1}} \left(1 + \Delta_{kink}^{n+1}\right) + I_{leak}^{n+1}. \tag{11}$$

Although not indicated here, the terms in (11), namely I_{sub}^{n+1}, I_a^{n+1}, Δ_{kink}^{n+1}, and I_{leak}^{n+1} depend on the voltage terms, $V_d^{n+\frac{1}{2}}$, $V_s^{n+\frac{1}{2}}$, and $V_g^{n+\frac{1}{2}}$. A detailed explanation of these terms will be shown in the next part of this section together with the explanation of the VinC LIM formulation for the drain-source current. For illustration, Fig. 3 shows the chain relationship between the terms in the version 1.0 RPI poly-Si TFT model from the terminal voltages, to the final drain-source current formula.

This method can also be extended to other RPI TFT models. For example, in the version 2.0 RPI poly-Si TFT model, the basic LIM current equation can be written as

$$I_{ds}^{n+1} = I_{ds1}^{n+1} \cdot \left(1 + \Delta_{kink}^{n+1}\right) + I_{leak}^{n+1} \tag{12}$$

where I_{ds1}^{n+1} is the channel current, and Δ_{kink}^{n+1} and I_{leak}^{n+1} are again the variables accounting for the kink effect and the leakage current respectively, and both of them are functions of $V_d^{n+\frac{1}{2}}$, $V_s^{n+\frac{1}{2}}$, and $V_g^{n+\frac{1}{2}}$. Compared to version 1.0, this version includes more effects and is more accurate for short-channel RPI TFT devices. The detailed formula for these terms will be shown in the next part of this section. Similar to Fig. 3, Fig. 4 shows the relationship between the voltage dependent variables in this version where more variables and more complex relationships can be observed.

B. VinC LIM FORMULATION FOR RPI TFT DRAIN-SOURCE CURRENT

To derive the VinC LIM formulation, the voltage equations at the drain, source and gate terminals of the TFT can first be written as

$$V_d^{n+1} = \frac{\frac{C_d}{\Delta t} V_d^n - \sum_{k=1,k\neq s}^{M_d} I_{dk}^{n+1} - I_{ds}^{n+1} + H_d^{n+1}}{\frac{C_d}{\Delta t} + G_d} \tag{13}$$

$$V_s^{n+1} = \frac{\frac{C_s}{\Delta t} V_s^n - \sum_{k=1,k\neq d}^{M_s} I_{sk}^{n+1} + I_{ds}^{n+1} + H_s^{n+1}}{\frac{C_s}{\Delta t} + G_s} \tag{14}$$

$$V_g^{n+1} = \frac{\frac{C_g}{\Delta t} V_g^n - \sum_{k=1}^{M_g} I_{gk}^{n+1} + H_g^{n+1}}{\frac{C_g}{\Delta t} + G_g} \tag{15}$$

where the same time index of $n + 1$ is used in the voltages and the currents, and the I_{ds} terms have been extracted out from the total current summations for emphasis. The signs of the I_{ds} currents are different in the two equations to signify a current direction flowing in or out of a node. The I_{ds} current is not shown in the gate voltage equation as in most cases, the gate node is independent of the I_{ds} current, unless the gate and drain, or gate and source nodes of a TFT is shorted, where either $V_g = V_d$, or $V_g = V_s$. In such cases, equation (13) or (14) will be used to represent the gate voltage equation depending on the respective situation.

Then, a general equation of I_{ds} can be written as

$$I_{ds} = I\left(V_d, V_g, V_s\right) \tag{16}$$

for a RPI TFT model where its drain-source current depends on the voltages at all the terminals of the TFT. Since the terminal voltages of the TFT can also be functions of I_{ds}, this equation can be further written as

$$I_{ds} = I\left(V_d\left(I_{ds}\right), V_g\left(I_{ds}\right), V_s\left(I_{ds}\right)\right) \tag{17}$$

where the time indices have been omitted to avoid convoluting the expressions.

To apply the concept of VinC LIM into the I_{ds} current formulation, (17) needs to be solved for the I_{ds} term. This can normally be done by moving all the I_{ds} terms in the right-hand side of the equation to the left-hand side of the equation,

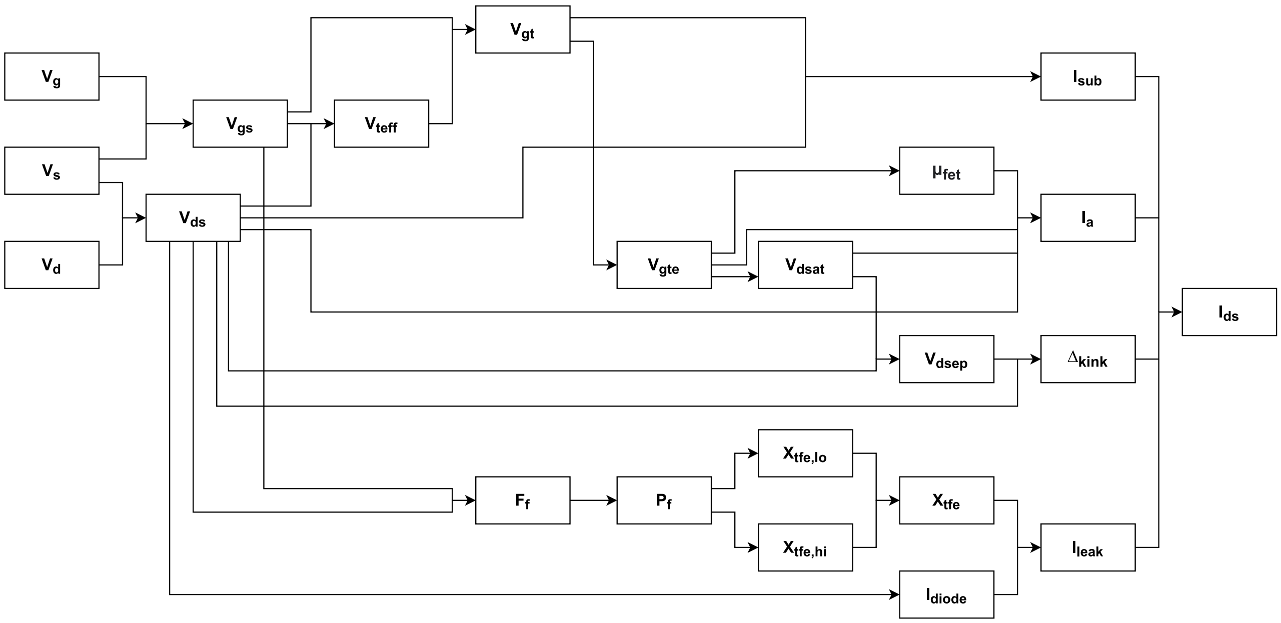


FIGURE 3. The chain relationship of the variables in version 1.0 RPI poly-Si TFT for a drain-source current simulation. All the variables will be discussed in Subsection III-B.

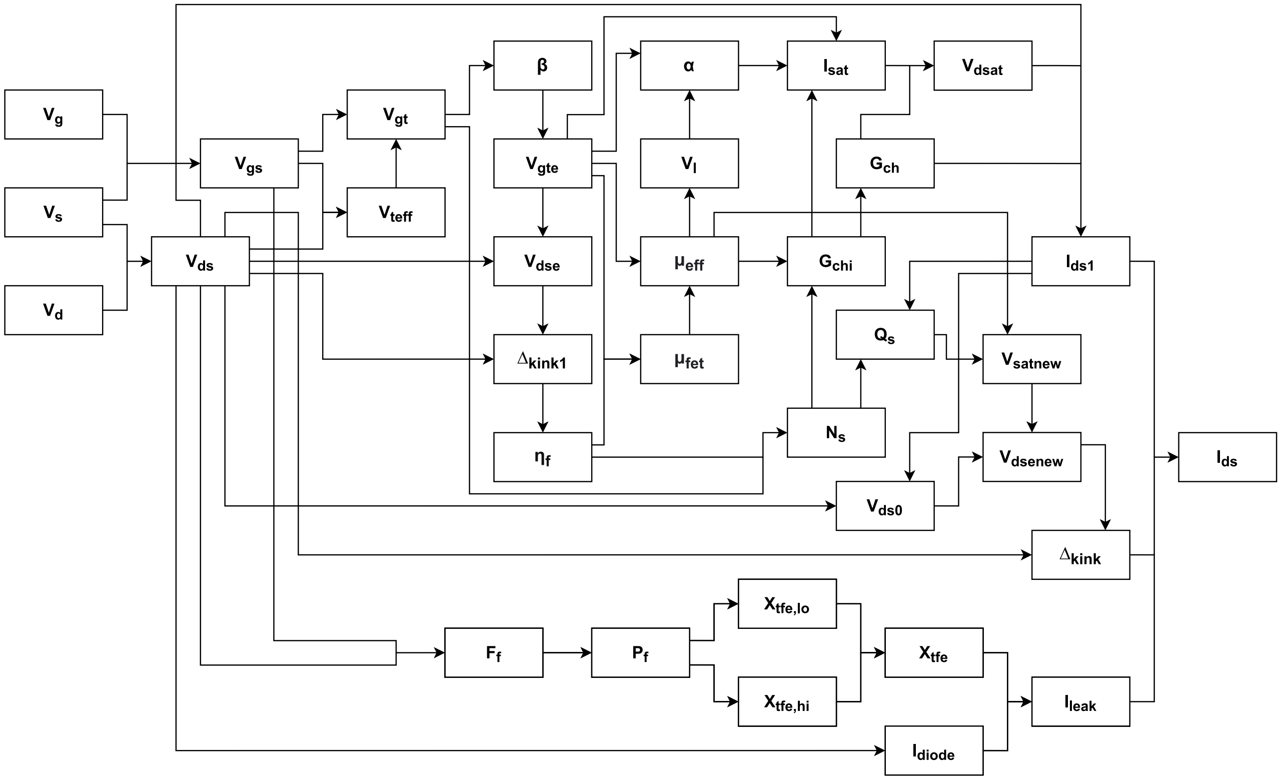


FIGURE 4. The chain relationship of the variables in version 2.0 RPI poly-Si TFT for a drain-source current simulation. All the variables will be discussed in Subsection III-B.

and solving for I_{ds}. However, since the TFT is a non-linear device, this is not a straightforward process, and linearizing it would cause accuracy drops especially when larger time step values are applied. To overcome this, a Newton-Raphson iteration is introduced to the process of solving the drain-source current. Such an idea has also been applied in the formulation of VinC LIM for diodes [25]. The general formula of the Newton-Raphson method is given by

$$x_{new} = x_{old} - \frac{F\left(x_{old}\right)}{F'\left(x_{old}\right)} \tag{18}$$

and by substituting the x terms into I_{ds}, the equation can be written as

$$I_{ds,new} = I_{ds,old} - \frac{F\left(I_{ds,old}\right)}{F'\left(I_{ds,old}\right)} \tag{19}$$

where $I_{ds,new}$ and $I_{ds,old}$ are the present and previous I_{ds} values in the Newton-Raphson loop, $F\left(I_{ds,old}\right)$ is the value of the differentiable function $F\left(I_{ds}\right)$ when substituted with the $I_{ds,old}$ value, and $F'\left(I_{ds,old}\right)$ is the derivative of $F\left(I_{ds,old}\right)$ with respect to $I_{ds,old}$. The differentiable equation $F\left(I_{ds}\right)$ and its derivative $F'\left(I_{ds}\right)$ can be obtained from (17), where they are described by

$$F\left(I_{ds}\right) = I_{ds} - I\left(V_d\left(I_{ds}\right), V_g\left(I_{ds}\right), V_s\left(I_{ds}\right)\right) = 0 \tag{20}$$

and

$$F'\left(I_{ds}\right) = 1 - I'\left(V'_d\left(I_{ds}\right), V'_g\left(I_{ds}\right), V'_s\left(I_{ds}\right)\right) = 0 \tag{21}$$

respectively, where $V'_d\left(I_{ds}\right) = -\left(\frac{C_d}{\Delta t} + G_d\right)^{-1}$ and $V'_s\left(I_{ds}\right) = \left(\frac{C_s}{\Delta t} + G_s\right)^{-1}$ are the resulting derivatives with respect to I_{ds} of the drain voltage, V_d, and the source voltage, V_s, from the LIM node equations in (13) and (14) respectively. Note that $V'_d\left(I_{ds}\right)$ and $V'_s\left(I_{ds}\right)$ can be zero if the node is independent of the I_{ds} current. The exact equations use in $F\left(I_{ds}\right)$ and $F'\left(I_{ds}\right)$ are dependent on the types and properties of the TFT models. In this formulation, the version 1.0 and version 2.0 RPI poly-Si TFT models are considered.

For the version 1.0 RPI poly-Si TFT with the I_{ds} equation shown in (10), the differentiable equation for this TFT model is given by

$$F\left(I_{ds}\right) = I_{ds} - \left(\frac{I_a \cdot I_{sub}}{I_a + I_{sub}}\left(1 + \Delta_{kink}\right) + I_{leak}\right) \tag{22}$$

and the derivative of the equation is given by

$$F'\left(I_{ds}\right) = 1 - \left(\frac{I'_a \cdot I^2_{sub} + I^2_a \cdot I'_{sub}}{\left(I_a + I_{sub}\right)^2} \cdot \left(1 + \Delta_{kink}\right) + \frac{I_a \cdot I_{sub}}{I_a + I_{sub}} \cdot \Delta'_{kink}\right) - I'_{leak} \tag{23}$$

where I'_a, I'_{sub}, Δ'_{kink}, and I'_{leak} are the derivatives of I_a, I_{sub}, Δ_{kink}, and I_{leak} respectively with respect to I_{ds}. Then from the RPI TFT modeling equations [26], the subthreshold current, I_{sub} is given by

$$I_{sub} = u_s \cdot f_{cox} \cdot \frac{w_{eff}}{l_{eff}} \cdot v^2_{sth} \cdot e^{\frac{V_{gt}}{v_{sth}}} \cdot \left(1 - e^{-\frac{V_{ds}}{v_{sth}}}\right) \tag{24}$$

where u_s is the subthreshold mobility, f_{cox} is the ratio of the gate insulator permittivity to the thin-oxide thickness, w_{eff} is the effective width, l_{eff} is the effective length, and $v_{sth} = \eta \cdot v_{th}$, where η is the subthreshold ideality factor and v_{th} is the thermal voltage at the device temperature. These four parameters are model parameters that are independent of I_{ds}. Conversely, the terms V_{gt} and V_{ds} are variables that depend on I_{ds}. V_{gt} can be expressed as

$$V_{gt} = V_{gs} - V_{teff} \tag{25}$$

with

$$V_{teff} = v_{to} - \frac{a_t \cdot V^2_{ds} + b_t}{l_{eff} \cdot \left(1 + e^{\frac{V_{gs} - v_{st} - v_{to}}{v_{si}}}\right)} \tag{26}$$

where a_t is the first drain induced barrier lowering (DIBL) parameter, b_t is the second DIBL parameter, v_{si} is the first parameter for V_{gs} dependence, v_{st} is the second parameter for V_{gs} dependence, v_{to} is the threshold voltage, and

$$V_{gs} = V_g - V_s. \tag{27}$$

The formula for V_{ds} is simply

$$V_{ds} = V_d - V_s. \tag{28}$$

Then, the derivatives of (24) to (28) with respect to I_{ds} are given by

$$I'_{sub} = u_s \cdot f_{cox} \cdot \frac{w_{eff}}{l_{eff}} \cdot v_{sth} \cdot e^{\frac{V_{gt}}{v_{sth}}} \cdot \left(V'_{gt} - e^{-\frac{V_{ds}}{v_{sth}}} \cdot \left(V'_{gt} - V'_{ds}\right)\right) \tag{29}$$

$$V'_{gt} = V'_{gs} - V'_{teff} \tag{30}$$

$$V'_{teff} = \frac{A' \cdot B - B' \cdot A}{l_{eff} \cdot B^2} \tag{31}$$

where $A = a_t \cdot V^2_{ds} + b_t, A' = 2 \cdot a_t \cdot V_{ds} \cdot V'_{ds}, B = 1 + e^{\frac{V_{gs} - v_{st} - v_{to}}{v_{si}}}$, and $B' = \frac{V'_{gs}}{v_{si}} \cdot e^{\frac{V_{gs} - v_{st} - v_{to}}{v_{si}}}$ are temporary variables used to simplify the equation,

$$V'_{gs} = V'_g - V'_s = -\left(\frac{C_s}{\Delta t} + G_s\right)^{-1} \tag{32}$$

by assuming that the gate terminal is independent of I_{ds}, and

$$V'_{ds} = V'_d - V'_s = -\left(\frac{C_d}{\Delta t} + G_d\right)^{-1} - \left(\frac{C_s}{\Delta t} + G_s\right)^{-1}. \tag{33}$$

For the above threshold current, I_a, the equation is given by

$$\begin{aligned} I_a &= \mu_{fet} \cdot f_{cox} \cdot \frac{w_{eff}}{l_{eff}} \cdot V_{ds} \cdot \left(V_{gte} - \frac{V_{ds}}{2 \cdot a_{sat}}\right), \\ V_{ds} &\le V_{dsat} \text{ (linear region)} \\ I_a &= \mu_{fet} \cdot f_{cox} \cdot \frac{w_{eff}}{l_{eff}} \cdot \frac{1}{2} \cdot V^2_{gte} \cdot a_{sat}, \\ V_{ds} &> V_{dsat} \text{ (saturation region)} \end{aligned} \tag{34}$$

where a_{sat} is the proportionality constant of V_{dsat}, and V_{dsat}, μ_{fet}, and V_{gte} are I_{ds} dependent variables with equations

$$V_{dsat} = a_{sat} \cdot V_{gte} \tag{35}$$

$$\mu_{fet} = \frac{u_0 \cdot u_1 \cdot \left(\frac{2 \cdot V_{gte}}{v_{sth}}\right)^{m_u}}{u_0 + u_1 \cdot \left(\frac{2 \cdot V_{gte}}{v_{sth}}\right)^{m_u}} \tag{36}$$

where u_0 and u_1 are the high and low field mobility parameters respectively, m_u is the low field mobility exponent applied in (36), and

$$V_{gte} = v_{sth} \cdot \left(1 + \frac{V_{gt}}{2 \cdot v_{sth}} + \sqrt{\delta^2 + \left(\frac{V_{gt}}{2 \cdot v_{sth}} - 1\right)^2}\right) \tag{37}$$

where δ is the transition width parameter. The differentiation of (34) to (37) then yields

$$\begin{aligned} I'_a = {} & f_{cox} \cdot \frac{w_{eff}}{l_{eff}} \cdot \left(\mu'_{fet} \cdot V_{ds} \cdot \left(V_{gte} - \frac{V_{ds}}{2 \cdot a_{sat}}\right) + \mu_{fet}\right. \\ & \cdot V'_{ds} \cdot \left(V_{gte} - \frac{V_{ds}}{2 \cdot a_{sat}}\right) \\ & \left. + \mu_{fet} \cdot V_{ds} \cdot \left(V'_{gte} - \frac{V'_{ds}}{2 \cdot a_{sat}}\right)\right), \\ & V_{ds} \leq V_{dsat} \text{ (linear region)} \\ I'_a = {} & f_{cox} \cdot \frac{w_{eff}}{l_{eff}} \cdot \frac{1}{2} \cdot a_{sat} \cdot V_{gte} \left(\mu'_{fet} \cdot V_{gte} + 2 \cdot V'_{gte} \cdot \mu_{fet}\right), \\ & V_{ds} > V_{dsat} \text{ (saturation region)} \end{aligned} \tag{38}$$

$$V'_{dsat} = a_{sat} \cdot V'_{gte} \tag{39}$$

$$\mu'_{fet} = \frac{u_0^2 \cdot u_1 \cdot m_u \cdot \left(\frac{2 \cdot V_{gte}}{v_{sth}}\right)^{m_u - 1} \cdot \frac{2 \cdot V'_{gte}}{v_{sth}}}{\left(u_0 + u_1 \cdot \left(\frac{2 \cdot V_{gte}}{v_{sth}}\right)^{m_u}\right)^2} \tag{40}$$

$$V'_{gte} = \frac{V'_{gt}}{2} \cdot \left(1 + \frac{\left(\frac{V_{gt}}{2 \cdot v_{sth} - 1}\right)}{\sqrt{\delta^2 + \left(\frac{V_{gt}}{2 \cdot v_{sth}} - 1\right)^2}}\right). \tag{41}$$

The kink effect is observed during large drain biasing in the TFT model and the equation is written as

$$\Delta_{kink} = a_{kinkt} \cdot \left(V_{ds} - V_{dsep}\right) \cdot e^{\frac{-v_{kink}}{V_{ds} - V_{dsep}}} \tag{42}$$

where $a_{kinkt} = \frac{1}{v_{kink}} \left(\frac{l_{kink}}{l_{eff}}\right)^{m_k}$, v_{kink} is the kink effect voltage, l_{kink} is the kink effect constant, m_k is the kink effect exponent, and V_{dsep} is the inner variable which depends on I_{ds}. It is given by

$$V_{dsep} = \frac{V_{ds}}{\left(1 + \left(\frac{V_{ds}}{V_{dsat}}\right)^3\right)^{\frac{1}{3}}} - v_{th} \tag{43}$$

where V_{dsat} is given in (35). The differentiation of (42) and (43) with respect to I_{ds} through the chain rule, yields

$$\begin{aligned} \Delta'_{kink} = a_{kinkt} \cdot \left(V'_{ds} - V'_{dsep}\right) & \\ \cdot e^{\frac{-v_{kink}}{V_{ds} - V_{dsep}}} \cdot \left(1 + \frac{v_{kink}}{V_{ds} - V_{dsep}}\right) & \end{aligned} \tag{44}$$

and

$$V'_{dsep} = \frac{V'_{ds} \cdot D - D' \cdot V_{ds}}{D^2} \tag{45}$$

where $D = \left(1 + \left(\frac{V_{ds}}{V_{dsat}}\right)^3\right)^{\frac{1}{3}}$ and $D' = \frac{V_{ds}^2}{V_{dsat}^4} (V'_{ds} \cdot V_{dsat} - V'_{dsat} \cdot V_{ds}) \left(1 + \left(\frac{V_{ds}}{V_{dsat}}\right)^3\right)^{-\frac{2}{3}}$ are temporary variables used to simplify the equation.

Finally, for the leakage current, I_{leak}, its effect to the overall drain-source current is minimal and its equation is described by

$$I_{leak} = i_0 \cdot w_{eff} \cdot \left(e^{\frac{b_{lk} \cdot V_{ds}}{v_{th}}} - 1\right) \left(X_{tfe} - x_{te}\right) + I_{diode} \tag{46}$$

where i_0 is the leakage scaling constant, b_{lk} is the leakage barrier lowering constant, x_{te} is a temperature-dependent variable, and X_{tfe} and I_{diode} are functions of I_{ds}, where X_{tfe} also depends on $X_{tfe,lo}$, $X_{tfe,hi}$, P_f, and F_f. Their detailed formulations can be referred from [26] and the relationship between the variables can be referred from Fig. 3. The derivative of (46) is given by

$$\begin{aligned} I'_{leak} = i_0 \cdot w_{eff} \cdot \left(\frac{b_{lk} \cdot V'_{ds}}{v_{th}} \cdot e^{\frac{b_{lk} \cdot V_{ds}}{v_{th}}} \cdot \left(X_{tfe} - x_{te}\right)\right. & \\ \left. + X'_{tfe} \cdot \left(e^{\frac{b_{lk} \cdot V_{ds}}{v_{th}}} - 1\right)\right) + I'_{diode}. & \end{aligned} \tag{47}$$

This completes the formulation for the drain current of the TFT in the version 1.0 RPI poly-Si TFT model. If intrinsic capacitances are not considered in the TFT model, then at this point, a complete VinC LIM simulation for the TFT circuit can be performed by using (19), (22) and (23) to solve for the I_{ds} currents and (13), (14) and (15) for the node voltages. To ease the convergence of the Newton-Raphson iterations, the I_{ds} value at the previous time step can be used as the initial guess in $I_{ds,old}$, and the converged value will be taken as the solution.

Next, the formulation for the drain-source current of version 2.0 of the RPI poly-Si TFT model will be discussed using the VinC LIM approach. Its drain-source current equation is given in (12) where the subthreshold current and above threshold current terms have been replaced with the channel current term. The differentiable equation and its derivative for this TFT model are described by

$$F\left(I_{ds}\right) = I_{ds} - \left(I_{ds1} \cdot \left(1 + \Delta_{kink}\right) + I_{leak}\right) \tag{48}$$

and

$$F'\left(I_{ds}\right) = 1 - \left(I'_{ds1} \cdot \left(1 + \Delta_{kink}\right) + I_{ds1} \cdot \Delta'_{kink}\right) - I'_{leak} \tag{49}$$

where I'_{ds1}, Δ'_{kink} and I'_{leak} are derivatives with respect to I_{ds} of I_{ds1}, Δ_{kink} and I_{leak}. The equation for the channel current, I_{ds1}, is given by

$$I_{ds1} = \frac{G_{ch} \cdot V_{ds} \cdot \left(1 + \lambda \cdot V_{ds}\right)}{\left(1 + \left(\frac{V_{ds}}{V_{dsat}}\right)^{m_e}\right)^{\frac{1}{m_e}}} \tag{50}$$

where λ is the channel length modulation parameter, m_e is the long channel saturation transition parameter, and G_{ch},

V_{dsat}, and V_{ds} are functions of I_{ds}. The derivative of (50) with respect to I_{ds} can be written as

$$I'_{ds1} = \frac{F' \cdot J - J' \cdot F}{J^2} \tag{51}$$

where

$$\begin{aligned}
F &= G_{ch} \cdot V_{ds} \cdot (1 + \lambda \cdot V_{ds}), \\
F' &= G'_{ch} \cdot V_{ds} \cdot (1 + \lambda \cdot V_{ds}) + G_{ch} \cdot V'_{ds} \cdot (1 + \lambda \cdot V_{ds}) \\
&\quad + G_{ch} \cdot V_{ds} \cdot \left(\lambda \cdot V'_{ds}\right), \\
J &= \left(1 + \left(\frac{V_{ds}}{V_{dsat}}\right)^{m_e}\right)^{\frac{1}{m_e}}, \text{ and} \\
J' &= \frac{V'_{ds} \cdot V_{dsat} - V_{ds} \cdot V'_{dsat}}{V_{dsat}^2} \cdot \left(\frac{V_{ds}}{V_{dsat}}\right)^{m_e - 1} \\
&\quad \cdot \left(1 + \left(\frac{V_{ds}}{V_{dsat}}\right)^{m_e}\right)^{\frac{1}{m_e} - 1}
\end{aligned}$$

are temporarily variables used to simplify the equation. V_{ds} and its derivative are given in (28) and (33) respectively, V_{dsat} and its derivative are given by

$$V_{dsat} = \frac{I_{sat}}{G_{ch}} \tag{52}$$

and

$$V'_{dsat} = \frac{I'_{sat} \cdot G_{ch} - G'_{ch} \cdot I_{sat}}{G_{ch}^2} \tag{53}$$

where G_{ch} and G'_{ch} are described as

$$G_{ch} = \frac{G_{chi}}{1 + G_{chi} \cdot (r_s + r_d)} \tag{54}$$

$$G'_{ch} = \frac{G'_{chi}}{(1 + G_{chi} \cdot (r_s + r_d))^2} \tag{55}$$

where r_s and r_d are the effective access resistances. Proceeding further, I_{sat} is given by

$$I_{\text{sat}} = \frac{G_{chi} \cdot V_{gte}}{G_{chi} \cdot r_s + 1 + \alpha + \sqrt{1 + 2 \cdot G_{chi} \cdot r_s + (1 + \alpha)^2}} \tag{56}$$

with G_{chi}, V_{gte}, and α are variables changing according to the terminal voltages and hence, depend on I_{ds}. Differentiating (56) with respect to I_{ds} through the chain rule yields

$$I'_{sat} = \frac{L' \cdot P - P' \cdot L}{P^2} \tag{57}$$

where $L = G_{chi} \cdot V_{gte}$, $L' = G'_{chi} \cdot V_{gte} + G_{chi} \cdot V'_{gte}$, $P = G_{chi} \cdot r_s + 1 + \alpha + \sqrt{1 + 2 \cdot G_{chi} \cdot r_s + (1 + \alpha)^2}$, and $P' = G'_{chi} \cdot r_s + \alpha' + \frac{G'_{chi} \cdot r_s + \alpha' \cdot (1 + \alpha)}{\sqrt{1 + 2 \cdot G_{chi} \cdot r_s + (1 + \alpha)^2}}$.
G_{chi} is given by

$$G_{chi} = q \cdot N_s \cdot \mu_{eff} \cdot \frac{w_{eff}}{l_{eff}} \tag{58}$$

where N_s and μ_{eff} are voltages dependent variables and q is the electron charge. The derivative of G_{chi} with respect to I_{ds} gives

$$G'_{chi} = q \cdot \frac{w_{eff}}{l_{eff}} \cdot \left(N'_s \cdot \mu_{eff} + N_s \cdot \mu'_{eff}\right). \tag{59}$$

For N_s and N'_s, the equations are given by

$$N_s = \frac{f_{cox} \cdot v_{sth}}{q} \cdot \ln\left(1 + \frac{1}{2} \cdot e^S\right) \tag{60}$$

$$N'_s = \frac{f_{cox} \cdot v_{sth}}{q} \cdot \frac{\frac{1}{2} \cdot S' \cdot e^S}{1 + \frac{1}{2} \cdot e^S} \tag{61}$$

where $S = \frac{V_{gt}}{\eta_f \cdot v_{th}}$ and $S' = \frac{V'_{gt} \cdot \eta_f - \eta'_f \cdot V_{gt}}{\eta_f^2 \cdot v_{th}}$.
α is given by

$$\alpha = \frac{V_{gte}}{V_l} \tag{62}$$

where $V_l = v_{max} \cdot \frac{l_{eff}}{\mu_{eff}}$ and v_{max} is the saturation velocity, and the derivative of α is given by

$$\alpha' = \frac{V'_{gte} \cdot V_l - V'_l \cdot V_{gte}}{V_l^2} \tag{63}$$

where $V'_l = -v_{max} \cdot \mu'_{eff} \cdot \frac{l_{eff}}{\mu_{eff}^2}$. For μ_{eff}, its expression is described as

$$\mu_{eff} = u_s + \frac{\mu_{fet}}{1 + \frac{\theta}{t_{ox}} \cdot V_{gte}} \tag{64}$$

where u_s is the subthreshold mobility, θ is the mobility degradation parameter, t_{ox} is the thin-oxide thickness, and μ_{fet} is a function of the voltages and hence the drain-source current. The differentiation of (64) with respect to I_{ds} results in

$$\mu'_{eff} = \frac{\mu'_{fet}\left(1 + \frac{\theta}{t_{ox}} \cdot V_{gte}\right) - \frac{\theta}{t_{ox}} \cdot V'_{gte} \cdot \mu_{fet}}{\left(1 + \frac{\theta}{t_{ox}} \cdot V_{gte}\right)^2}. \tag{65}$$

Then, the equations for μ_{fet} and μ'_{fet} are given by

$$\mu_{fet} = \frac{u_0 \cdot u_1 \cdot \left(\frac{2 \cdot V_{gte}}{\eta_f \cdot v_{th}}\right)^{m_u}}{u_0 + u_1 \cdot \left(\frac{2 \cdot V_{gte}}{\eta_f \cdot v_{th}}\right)^{m_u}} \tag{66}$$

and

$$\mu'_{fet} = \frac{\begin{array}{c} u_0^2 \cdot u_1 \cdot m_u \cdot \frac{2}{v_{th}} \\ \cdot \frac{V'_{gte} \cdot \eta_f - \eta'_f \cdot V_{gte}}{\eta_f^2} \cdot \left(\frac{2 \cdot V_{gte}}{\eta_f \cdot v_{th}}\right)^{m_u - 1} \end{array}}{\left(u_0 + u_1 \cdot \left(\frac{2 \cdot V_{gte}}{\eta_f \cdot v_{th}}\right)^{m_u}\right)^2} \tag{67}$$

where all the model parameters and variables have been introduced above. Next, the expression for η_f is given by

$$\eta_f = \frac{\eta}{1 + m_\eta \cdot \frac{\eta - 1}{\eta} \cdot \frac{\Delta_{kink1}}{1 + \Delta_{kink1}}} \tag{68}$$

where m_η is the η floating-body parameter, and Δ_{kink1} is similar to the equation described in (42) but by changing V_{dsep} to V_{dse} and multiplying by the ratio of w_{eff}/l_{eff}. The derivative of (68) is given by

$$\eta_f' = \frac{-m_\eta \cdot (\eta - 1) \cdot \frac{\Delta_{kink1}'}{(1+\Delta_{kink1})^2}}{\left(1 + m_\eta \cdot \frac{\eta-1}{\eta} \cdot \frac{\Delta_{kink1}}{1+\Delta_{kink1}}\right)^2} \tag{69}$$

where Δ_{kink1}' is also similar to (44), but by changing the V_{dsep} and V_{dsep}' terms to V_{dse} and V_{dse}' respectively and multiplying by the ratio of w_{eff}/l_{eff}. V_{dse} is expressed as

$$V_{dse} = \frac{V_{ds}}{\left(1 + \left(\frac{V_{ds}}{V_{sat}}\right)^{m_{ss}}\right)^{\frac{1}{m_{ss}}}} - v_{th} \tag{70}$$

and its derivative is given by

$$V_{dse}' = \frac{V_{ds}' \cdot T - T' \cdot V_{ds}}{T^2} \tag{71}$$

by assuming $T = \left(1 + \left(\frac{V_{ds}}{V_{sat}}\right)^{m_{ss}}\right)^{\frac{1}{m_{ss}}}$ and $T' = \frac{(V_{ds}' \cdot V_{sat} - V_{sat}' \cdot V_{ds})}{V_{sat}^2} \cdot \left(\frac{V_{ds}}{V_{sat}}\right)^{m_{ss}-1} \cdot \left(1 + \left(\frac{V_{ds}}{V_{sat}}\right)^{m_{ss}}\right)^{\frac{1}{m_{ss}}-1}$ where m_{ss} is the V_{dse} transition parameter, $V_{sat} = V_{gte}$, and hence, $V_{sat}' = V_{gte}'$. In conjunction with this,

$$V_{gte} = v_{sth} \cdot \left(1 + \beta + \sqrt{\delta^2 + (\beta - 1)^2}\right) \tag{72}$$

and

$$V_{gte}' = v_{sth} \cdot \beta' \cdot \left(1 + \frac{\beta - 1}{\sqrt{\delta^2 + (\beta - 1)^2}}\right) \tag{73}$$

where $\beta = a_{sat} \cdot \frac{V_{gt}}{2 \cdot v_{sth}}$ and $\beta' = a_{sat} \cdot \frac{V_{gt}'}{2 \cdot v_{sth}}$. The expressions for V_{gt} and V_{gt}' can be referred from (25) to (28) and (30) to (33) as they are the same in both version 1.0 and 2.0 RPI poly-Si TFT.

The formula for the kink effect in the version 2.0 model is written as

$$\Delta_{kink} = a_{kinkt} \cdot (V_{ds} - V_{dsenew}) \cdot e^{\frac{-v_{kink}}{V_{ds} - V_{dsenew}}} \tag{74}$$

where V_{dsenew} is a function of I_{ds} and is similar to (70) by substituting V_{ds} with V_{ds0} and V_{sat} with V_{satnew}. Differentiating Δ_{kink} with respect to I_{ds} yields

$$\Delta_{kink}' = a_{kinkt} \cdot \left(V_{ds}' - V_{dsenew}'\right) \cdot e^{\frac{-v_{kink}}{V_{ds} - V_{dsenew}}} \cdot \left(1 + \frac{v_{kink}}{V_{ds} - V_{dsenew}}\right) \tag{75}$$

where V_{dsenew}' is similar to (71) by changing V_{ds}' to V_{ds0}' and V_{sat}' to V_{satnew}'. The equations for V_{ds0} and V_{ds0}' are given by

$$V_{ds0} = V_{ds} - I_{ds1} \cdot (r_s + r_d) \tag{76}$$

and

$$V_{ds0}' = V_{ds}' - I_{ds1}' \cdot (r_s + r_d) \tag{77}$$

while the equations for V_{satnew} and V_{satnew}' are given by

$$V_{satnew} = \frac{2 \cdot v_{max} \cdot l_{eff} \cdot Q_s}{Q_s \cdot \mu_{eff} + 2 \cdot v_{max} \cdot l_{eff} \cdot \frac{f_{cox}}{a_{sat}}} \tag{78}$$

and

$$V_{satnew}' = 2 \cdot v_{max} \cdot l_{eff} \cdot \frac{\begin{pmatrix} Q_s' \cdot \left(2 \cdot v_{max} \cdot l_{eff} \cdot \frac{f_{cox}}{a_{sat}}\right) \\ -Q_s^2 \cdot \mu_{eff}' \end{pmatrix}}{\left(Q_s \cdot \mu_{eff} + 2 \cdot v_{max} \cdot l_{eff} \cdot \frac{f_{cox}}{a_{sat}}\right)^2}. \tag{79}$$

where the variable Q_s and its derivative are defined as

$$Q_s = q \cdot N_s - I_{ds1} \cdot f_{cox} \cdot r_d \tag{80}$$

and

$$Q_s' = q \cdot N_s' - I_{ds1}' \cdot f_{cox} \cdot r_d. \tag{81}$$

Finally, the formula for the leakage current and its derivative are similar to (46) and (47). The exact relationship between the variables that make up the leakage current can be referred from [26] and Fig. 4.

This completes the formulation for the drain current of the TFT in the version 2.0 RPI poly-Si TFT model. A VinC LIM simulation for the TFT circuit in the model can be done similar to that in version 1.0, but by using the appropriate equations in (48) to (81) for the version 2.0 model.

Do note that all the TFT equations presented thus far are for the n-type TFT model. For a p-type TFT model, the polarity of the voltage values should be inverted and the current value calculated is flipped in its polarity as well at the end of its calculation. As an illustration, the current equation for a p-type TFT model in basic LIM is given as

$$I_{ds} = -I\left(-V_d, -V_g, -V_s\right) \tag{82}$$

while in the VinC LIM formulation it is given as

$$F\left(I_{ds}\right) = -I_{ds} - I\left(-V_d\left(I_{ds}\right), -V_g\left(I_{ds}\right), -V_s\left(I_{ds}\right)\right) = 0 \tag{83}$$

$$F'\left(I_{ds}\right) = -1 - I'\left(-V_d'\left(I_{ds}\right), -V_g'\left(I_{ds}\right), -V_s'\left(I_{ds}\right)\right) = 0 \tag{84}$$

which are then used in (19).

This completes the formulation for the drain-source current for TFTs in both the basic LIM and VinC LIM formulations. The effects of the intrinsic capacitance and its current formulation will be discussed next.

C. BASIC LIM AND VinC LIM FORMULATIONS FOR RPI TFT INTRINSIC CAPACITANCE CURRENTS

While the main current in a TFT is the drain-source current discussed in the previous section, there are also minor current flows associated with the gate-drain terminals and gate-source terminals which can affect the accuracy of the simulation. These currents are modeled by intrinsic capacitances

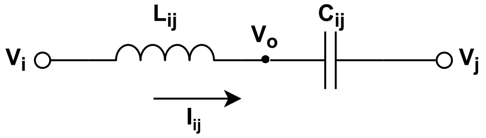


FIGURE 5. The branch topology for branch capacitance simulation in basic LIM.

which exist between the two terminals, as shown in Fig. 2, and are contributed by the overlap capacitances which are formed according to the physical properties such as the thin-oxide thickness, permittivity of the gate insulator, fringing factor, width of the model, etc., and the parasitic capacitances which depend on the model used for the TFT. These non-zero capacitances, labelled as C_{gd} and C_{gs}, connect the terminals of the TFT and hence, currents, labelled as I_{gd} and I_{gs}, are able to flow across the terminals when there are changes in the terminal potentials. In this section, the basic LIM and VinC LIM approach for these intrinsic capacitance currents are discussed.

The intrinsic capacitances in a TFT can be represented as a branch capacitor connection in the LIM topology. In the basic LIM formulation, a fictitious inductor is inserted into the branch with the capacitor. This is illustrated in Fig. 5, where L_{ij} is the inserted latency to the branch and V_o is the voltage at the node between the branch capacitor and the fictitious inductor. Based on this, it can be written that

$$V_c^{n+\frac{1}{2}} = V_c^{n-\frac{1}{2}} + \frac{\Delta t}{C_{ij}} I_{ij}^n \tag{85}$$

where $V_c = V_o - V_j$ is the voltage drop across the branch capacitor and

$$I_{ij}^{n+1} = I_{ij}^n + \frac{\Delta t}{L_{ij}} V_L^{n+\frac{1}{2}} \tag{86}$$

is the current flowing in the branch, where $V_L^{n+\frac{1}{2}}$ can be further expressed as

$$V_L^{n+\frac{1}{2}} = V_i^{n+\frac{1}{2}} - V_j^{n+\frac{1}{2}} - V_c^{n+\frac{1}{2}}. \tag{87}$$

Substituting (87) into (86) yields

$$I_{ij}^{n+1} = I_{ij}^n + \frac{\Delta t}{L_{ij}} \left(V_i^{n+\frac{1}{2}} - V_j^{n+\frac{1}{2}} - V_c^{n+\frac{1}{2}} \right) \tag{88}$$

which is the branch capacitance current updating equation for basic LIM. Thus, in a basic LIM simulation, (85) is applied first followed by (88) to solve for the branch capacitor current. In a TFT model, i and j can simply be replaced by the respective terminal nodes (e.g. g, d or s), and the value of C_{ij} can be obtained from the model. The drawback of this approach is the additional latency component that is inserted to the respective branch, which may affect the simulation accuracy and stability.

For the VinC LIM formulation, the current equation for the branch capacitor can be derived by substituting the respective voltage terms with their LIM voltage equations and solving for the I_{ij}^{n+1} current. Note that the insertion of fictitious inductances are not required for this approach. The branch current VinC LIM equation is given by

$$I_{ij}^{n+1} = \frac{\begin{array}{c} K_i \left(\frac{C_i V_i^n}{\Delta t} + H_i^{n+1} - \sum_{k=1,k\neq j}^{M_i} I_{ik}^{n+1} \right) - V_i^n \\ -K_j \left(\frac{C_j V_j^n}{\Delta t} + H_j^{n+1} - \sum_{k=1,k\neq i}^{M_j} I_{jk}^{n+1} \right) + V_j^n \end{array}}{\frac{\Delta t}{C_{ij}} + K_i + K_j} \tag{89}$$

where $K_i = \left(G_i + \frac{C_i}{\Delta t} \right)^{-1}$, $K_j = \left(G_j + \frac{C_j}{\Delta t} \right)^{-1}$ and the other terms are as described in Section II. Equation (89) is an implicit VinC LIM branch capacitance current formulation. The original paper showing the complete derivation steps for a semi-implicit VinC LIM branch capacitance formulation can be referred to in [27].

Equation (89) can be used to solve for the intrinsic capacitance currents in a TFT by substituting i and j with the respective terminal nodes (e.g. g, d or s) and the value of C_{ij} can be obtained from the model. It can be noted that the equation can be simplified into

$$I_{gd/gs}^{n+1} = \frac{\begin{array}{l} V_g^{n+1} - V_g^n + V_{d/s}^n \\ \quad -K_{d/s} \left(\frac{C_{d/s} V_{d/s}^n}{\Delta t} + H_{d/s}^{n+1} - \sum_{k=1,k\neq g}^{M_{d/s}} I_{dk/sk}^{n+1} \right) \end{array}}{\frac{\Delta t}{C_{gd/gs}} + K_{d/s}} \tag{90}$$

in a situation where terminal g is connected to a voltage source, or

$$I_{gd/gs}^{n+1} = \frac{\begin{array}{c} K_g \left(\frac{C_g V_g^n}{\Delta t} + H_g^{n+1} - \sum_{k=1,k\neq d/s}^{Mg} I_{gk}^{n+1} \right) \\ -V_g^n - V_{d/s}^{n+1} + V_{d/s}^n \end{array}}{\frac{\Delta t}{C_{gd/gs}} + K_g} \tag{91}$$

in a situation where terminal d or s is connected to a voltage source.

Considering a RPI poly-Si TFT model (for both versions 1.0 and 2.0), the overlap capacitances are given by

$$c_{gsos} = c_{gso} \cdot w_{eff_{overlap}} \tag{92}$$

$$c_{gdos} = c_{gdo} \cdot w_{eff_{overlap}} \tag{93}$$

where c_{gsos} is the gate-source overlap capacitance, c_{gdos} is the gate-drain overlap capacitance, $w_{eff_{overlap}}$ is the effective overlap width which depends on the area calculation method used in the TFT model, and c_{gso} and c_{gdo} are the source overlap capacitance factor and the drain overlap capacitance factor respectively, which can be calculated from the formula

$$c_{gso} = c_{gdo} = (l_f + l_d) \cdot \frac{\varepsilon_{gate}}{t_{ox}} \tag{94}$$

where l_f, l_d, t_{ox}, and ε_{gate} are the fringing factor, lateral diffusion into channel from source and drain, thin-oxide thickness, and permittivity of the gate insulator respectively.

Then, for the parasitic capacitance, it is evaluated depending on the selected capacitance model. For example, a constant capacitance model which only depends on the device parameters can be calculated from the model equation as

$$C_{gsp} = C_{gdp} = \frac{w_{eff} \cdot l_{eff} \cdot f_{cox}}{2}. \tag{95}$$

The total intrinsic capacitance of the RPI poly-Si TFT model is the summation of the overlap capacitance and the parasitic capacitance and can be written as

$$C_{gs} = c_{gsos} + C_{gsp} \tag{96}$$

and

$$C_{gd} = c_{gdos} + C_{gdp} \tag{97}$$

where C_{gs} is the gate-source intrinsic capacitance and C_{gd} is the gate-drain intrinsic capacitance.

D. PRACTICAL STEPS FOR ACCURACY IMPROVEMENT IN VinC LIM

One of the main differences between the basic LIM and VinC LIM algorithms is the utilization of a forward branch-marching scheme when solving for the branch currents in VinC LIM. Unlike the basic LIM branch equation that depends only on the voltages at the nodes and the current through that branch at the previous time step, the VinC branch equation depends also on the currents through the other branches which are solved for at the same time step. The forward branch-marching scheme alleviates this by using the most recently available current based on the order in which the currents are solved. However, this creates a situation where the solution is dependent on the order in which the branches are stepped through in the simulation. In a physical simulation, this order can be selected to start from the voltage and current sources, and propagating outwards to the rest of the circuit. However, determining this order is not a trivial task. To reduce the reliance on any particular order, in this work, a random order is used, but the order is alternated each time the branches are evaluated. In other words, if initially the order is selected as I_a to I_z, then on the next time step, the order will be flipped to be I_z to I_a. This idea is similar to other alternating direction algorithms, most notably to the one presented in [7] for transmission lines.

Besides that, both basic LIM and VinC LIM are still inherently explicit methods, where the solution of the current time point is based on the solution at the previous time point. While this avoids the solution of large systems of simultaneous equations, such as those present in SPICE, the accuracy of the solution depends on the size of the time step used. In basic LIM, this is less of an issue, since the maximum time step to ensure stability is normally much less than that which is needed to obtain an accurate result, but since VinC LIM relaxes the stability criteria, accuracy can degrade when significantly larger time steps are used. In our experiments, we found that the accuracy can be improved by repeating the branch evaluations to obtain a better convergence during the forward branch-marching scheme. This can provide a better trade-off between speed and accuracy, compared to using a smaller time step, since only part of the solution process needs to be repeated. Numerical examples are presented in Section IV which illustrate this trade-off.

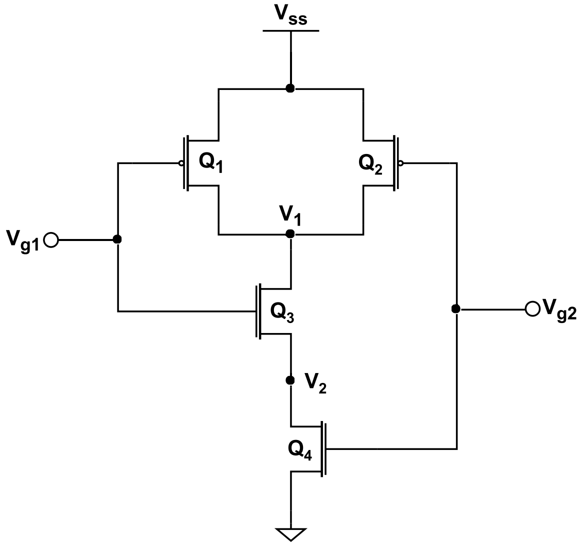


FIGURE 6. NAND circuit using TFT devices.

IV. RESULTS AND DISCUSSION

In this section, numerical examples are presented which show the application of the developed formulations in the simulations of TFT circuits. Silvaco's SmartSpice^{TM} will be used as a benchmark. All simulations are performed on a Linux server with an Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30 GHz with 528 GB of RAM.

A. NAND CIRCUIT

In the first example, a NAND circuit is constructed using TFTs modeled with the version 1.0 RPI poly-Si model. This example is simulated as a verification of the basic LIM and VinC LIM formulations presented in the previous section. Fig. 6 shows the circuit schematic for this example, where Q_1 to Q_4 are TFTs, $V_{ss} = 5V$ is the power supply voltage, V_{g1} and V_{g2} are the input signals, and V_1 and V_2 are the output nodes of the circuit. In this example, the intrinsic capacitance currents are ignored and only the drain-source current of each TFT in the circuit is considered.

Fig. 7 shows the simulated waveforms obtained from SmartSpice, basic LIM, and VinC LIM at the input and output nodes for the circuit. Fictitious capacitances of 10^{-18} F are inserted at the two LIM nodes of the circuit for the LIM simulations. The simulation using basic LIM is performed at a time step of 1 ps, which is its maximum stable time step, while the simulation using VinC LIM can be performed at a timestep of 50 ps which is 50× the maximum stable time step in basic LIM, while still being both stable and accurate. This clearly shows the advantage of the VinC LIM formulation over the basic LIM formulation. All simulations are comparable in terms of accuracy to SmartSpice.

B. 7T1C PIXEL CELL CIRCUIT

In the second example, a single 7T1C cell, which makes up one of the color cells in an RGB pixel of an OLED display

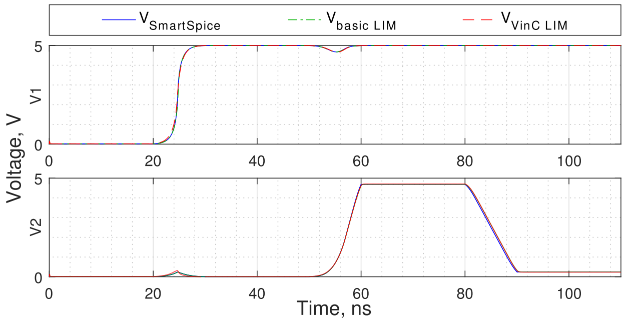


FIGURE 7. Version 1.0 RPI poly-Si TFT NAND circuit simulation results for SmartSpice, basic LIM (Δt = 1 ps), and VinC LIM (Δt = 50 ps) at nodes V_1 and V_2.

is simulated. The circuit diagram of the 7T1C cell is shown in Fig. 8 where it consists of 7 TFTs, 6 capacitors, 8 resistors and a diode. Fictitious latency components are inserted at the nodes and branches for the LIM simulations. Existing LIM formulations for diodes and branch capacitors are used and they can be referred from [25] and [27].

Two simulations are performed using this circuit. In the first simulation, a version 2.0 RPI poly-Si model is used for the drain-source current and the intrinsic capacitance currents are ignored. Fig. 9 shows the results obtained from SmartSpice, basic LIM, and VinC LIM at three distinct nodes in the circuit, V_1, V_2, and V_3. In addition, Table 1 tabulates the number of time points, runtime, and RMS error in the simulations for different time step values for basic LIM and VinC LIM, based on the maximum stable time step in basic LIM, which is 0.224 ps. Note that some additional breakpoints are added by the simulator in order to capture the changing edges of the input waveforms accurately in all simulations.

TABLE 1. Total number of time points, runtime, and RMS error in basic LIM and VinC LIM for different time step values for the 7T1C cell circuit without intrinsic capacitance.

Time step	Number of time points	Runtime (s)		Speedup (vs. basic LIM)	RMS error (vs. basic LIM)
		Basic LIM	VinC LIM		
1×	447,215	7.81	14.2	0.55×	0.000045
2×	223,608	unstable	7.17	1.09×	0.000280
10×	44,723	unstable	1.45	5.39×	0.001742
50×	8,945	unstable	0.27	28.9×	0.010369
100×	4,473	unstable	0.14	55.8×	0.020620
200×	2,237	unstable	0.07	112×	0.044238

From Fig. 9 and Table 1, the result from basic LIM is only stable at its maximum stable time step value which is 0.224 ps for this circuit. On the other hand, the VinC LIM simulations remain stable even though the time step is increased up to 200 times the initial value. Comparing the runtime, for the same time step, basic LIM is faster than VinC LIM, due to the simplicity of its formulation. However, as VinC LIM is not limited by the stability criteria of basic LIM, it greatly outperforms basic LIM on larger time steps. For example, a VinC LIM simulation with a time step of

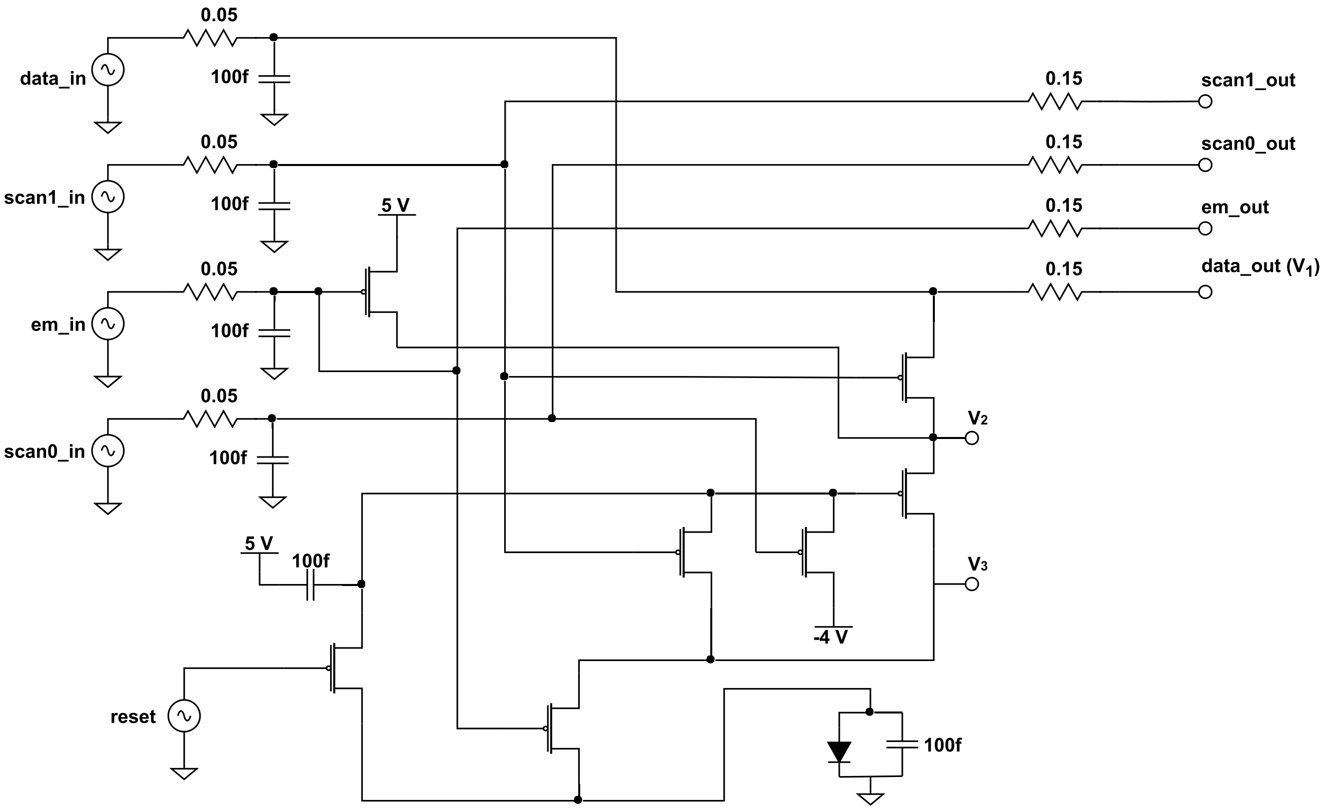


FIGURE 8. 7T1C pixel cell circuit.

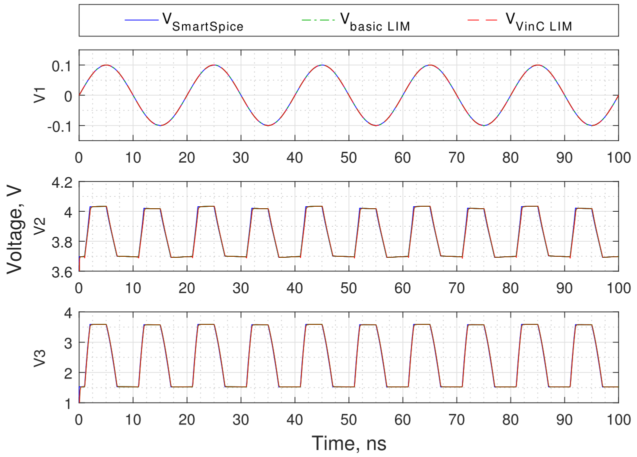


FIGURE 9. Version 2.0 RPI poly-Si TFT 7T1C cell circuit (without intrinsic capacitance) simulation results for SmartSpice, basic LIM ($\Delta t = \Delta t_{max} = 0.224$ ps), and VinC LIM ($\Delta t = 50 \times \Delta t_{max}$) at nodes V_1, V_2, and V_3.

TABLE 2. Total number of time points, runtime, and RMS error in basic LIM and VinC LIM under different conditions for the 7T1C cell circuit with intrinsic capacitance. Note that *bl* represents the number of branch loops that are performed.

Time step	Number of time points	Runtime (s)				RMS error (vs. basic LIM) $\times 10^{-2}$		
		Basic LIM	VinC LIM					
			1× *bl*	5× *bl*	10× *bl*	1× *bl*	5× *bl*	10× *bl*
1×	447,215	7.86	14.5	-	-	0.40	-	-
10×	44,723	unstable	1.45	-	-	1.8	-	-
25×	17,889	unstable	0.56	2.34	-	4.5	0.92	-
50×	8,945	unstable	0.29	1.11	2.22	9.2	1.8	0.83
100×	4,473	unstable	0.14	0.58	1.11	18	3.5	1.5

50× the maximum stable time step in basic LIM is 28.9 times faster, while retaining the accuracy. If simulation speed is the sole consideration in a simulation, and slight deviations in accuracy are acceptable, then larger time steps can also be used, for example 100× or 200× as shown in the table.

Next, the same simulation using this circuit is repeated, but with the intrinsic capacitance currents taken into account. Fig. 10 shows the simulation results from SmartSpice, basic LIM and VinC LIM. Additional branch loops are performed in VinC LIM as described in Section III-D to improve the accuracy. Table 2 shows a comparison of the runtime and accuracy of basic LIM and VinC LIM under different conditions. It is observed that the VinC LIM simulations are able to outperform basic LIM in terms of runtime at larger time steps, but with some degradation in accuracy. By adjusting the number of branch loops, the accuracy of VinC LIM can be improved, while still retaining its runtime advantage over basic LIM.

C. FULL TFT FLAT-PANEL DISPLAY CIRCUIT

In this third example, full TFT flat-panel display (FPD) circuits are simulated in VinC LIM and SmartSpice, to show the advantage of VinC LIM on large circuits compared to state-of-the-art commercial simulators. First, a full RGB pixel cell is simulated, where it consists of three 7T1C cells, each representing the red, green, and blue color in a pixel. A version 2.0 RPI poly-Si model is used for the drain-source current and the intrinsic capacitance currents are all taken into account. Fig. 11 shows the results from SmartSpice and VinC LIM where a similar level of accuracy can be observed. It is to be noted that the simulation in VinC LIM was carried out using a time step 5000 times the maximum stable time step of basic LIM without any extra loop on the branch simulations. If necessary, the simulation speed can be further boosted by using a larger time step and a proper control on the number of branch loops.

Then, large FPD circuits are constructed by using the RGB pixel cell to form the appropriate resolutions as in Table 3. Each circuit is simulated in VinC LIM and SmartSpice and the time spent per iteration is compared. It can be seen that VinC LIM greatly outperforms SPICE based simulators such as SmartSpice, especially on larger circuits. The runtime of VinC LIM scales almost linearly with the number of nodes in the circuit, while SPICE and its reliance on matrix based operations start to show very poor scaling on very large circuits. For circuits with more than 10 million nodes or devices, SmartSpice simulations were either unable to be

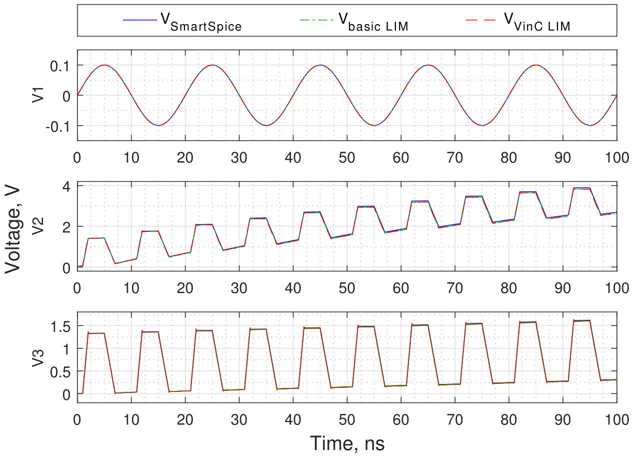


FIGURE 10. Version 2.0 RPI poly-Si TFT 7T1C cell circuit (with intrinsic capacitance) simulation results for SmartSpice, basic LIM ($\Delta t = \Delta t_{max} = 0.224$ ps), and VinC LIM ($\Delta t = 100 \times \Delta t_{max}$, $10 \times bl$) at nodes V_1, V_2, and V_3.

TABLE 3. Time spent per iteration for SmartSpice and VinC LIM in full TFT FPD circuits.

Circuit size (in pixels)	No. of nodes	No. of TFTs	Time per iteration (s)		Speedup ratio
			Smart-Spice	VinC LIM	
20×12	5,867	5,040	0.032	0.015	2.13×
25×20	12,159	10,500	0.080	0.030	2.67×
80×36	69,479	60,480	0.803	0.182	4.41×
100×100	240,851	210,000	6.362	0.677	9.40×
320×180	1,383,915	1,209,600	256.51	4.118	62.29×
640×360	5,532,627	4,838,400	4393.41	18.83	233.32×
960×540	12,446,147	10,886,400	dnc.	43.78	-
1920×1080	49,775,555	43,545,600	dnc.	193.30	-

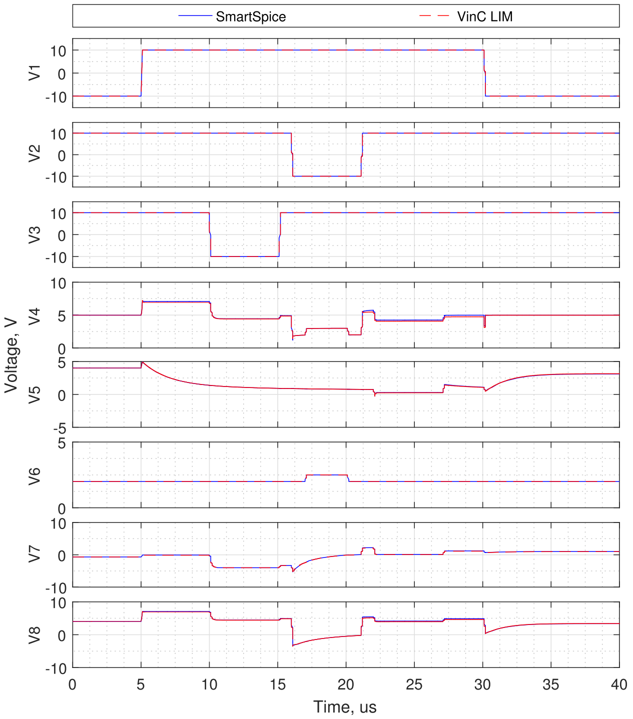


FIGURE 11. Version 2.0 RPI poly-Si TFT RGB circuit (with intrinsic capacitance) simulation results for SmartSpice and VinC LIM ($\Delta t = 1.12n = 5000 \times \Delta t_{max}$) at eight different nodes of a full RGB pixel cell.

completed even after seven days of runtime, or exceeded the total memory available on the systems. This is indicated by an entry of "dnc." in the table.

V. CONCLUSION

In this work, formulations for TFTs have been presented in the basic LIM and VinC LIM algorithms. The equations for the drain-source currents of different versions of the RPI TFT are reconstructed to work with the LIM algorithms. Besides that, the intrinsic capacitance currents flowing internally in a TFT device are also taken into consideration in the LIM formulations presented. Results show the superiority of the VinC LIM formulation in terms of stability compared to the basic LIM formulation, where the time steps in VinC LIM can be 50-5000 times larger than the time steps in basic LIM depending on the circuit simulated. In addition, on large circuits, VinC LIM greatly outperforms conventional SPICE-based simulators in terms of runtime.

Given the potential shown by VinC LIM in these results, a number of future works are currently being pursued. This includes the incorporation of secondary effects such as temperature, scaling, and self-heating in the simulations, and also on the parallelization of the VinC LIM algorithm for further runtime improvements on multithreaded operations. With these advancements in LIM, they can open up new choices in selecting circuit simulators in heavy computing scenarios.

ACKNOWLEDGMENT

The authors would like to thank Kostas Malinauskas and Alexander Samoylov from Silvaco Inc., Santa Clara, CA, USA, for setting up the data structures and the testing and quality assurance routines, used in this work.

REFERENCES

[1] L. W. Nagel and D. Pederson, "SPICE (simulation program with integrated circuit emphasis)," Dept. EECS, Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/ERL M382, Apr. 1973.

[2] J. E. Schutt-Ainé, "Latency insertion method (LIM) for the fast transient simulation of large networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 1, pp. 81–89, Jan. 2001.

[3] T. Sekine and H. Asai, "Block-latency insertion method (Block-LIM) for fast transient simulation of tightly coupled transmission lines," *IEEE Trans. Electromagn. Compat.*, vol. 53, no. 1, pp. 193–201, Feb. 2011.

[4] P. Goh, J. E. Schutt-Aine, D. Klokotov, J. Tan, P. Liu, W. Dai, and F. Al-Hawari, "Partitioned latency insertion method with a generalized stability criteria," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 1, no. 9, pp. 1447–1455, Sep. 2011.

[5] H. Kurobe, T. Sekine, and H. Asai, "Predictor-corrector latency insertion method for fast transient analysis of ill-constructed circuits," in *Proc. 17th Asia South Pacific Design Autom. Conf.*, Jan. 2012, pp. 365–370.

[6] H. Kurobe, T. Sekine, and H. Asai, "Locally implicit LIM for the simulation of PDN modeled by triangular meshes," *IEEE Microw. Wireless Compon. Lett.*, vol. 22, no. 6, pp. 291–293, Jun. 2012.

[7] H. Kurobe, T. Sekine, and H. Asai, "Alternating direction explicit-latency insertion method (ADE-LIM) for the fast transient simulation of transmission lines," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 2, no. 5, pp. 783–792, May 2012.

[8] K. H. Tan, P. Goh, and M. F. Ain, "Voltage-in-current formulation for the latency insertion method for improved stability," *Electron. Lett.*, vol. 52, no. 23, pp. 1904–1906, Nov. 2016.

[9] S. N. Lalgudi, M. Swaminathan, and Y. Kretchmer, "On-chip power-grid simulation using latency insertion method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 3, pp. 914–931, Apr. 2008.

[10] D. Klokotov, P. Goh, and J. E. Schutt-Ainé, "Latency insertion method (LIM) for DC analysis of power supply networks," *IEEE Trans. Compon., Package., Manuf. Technol.*, vol. 1, no. 11, pp. 1839–1845, Nov. 2011.

[11] T. Sekine and H. Asai, "Efficient modeling and analysis of multilayered power distribution network by using conformal mesh and block-type leapfrog scheme," in *Proc. IEEE Electr. Design Adv. Packag. Syst. Symp. (EDAPS)*, Dec. 2013, pp. 221–224.

[12] S. Okada, T. Sekine, and H. Asai, "Multi-rate locally implicit block leapfrog scheme for fast transient analysis of multi-layered power/ground planes," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 6, pp. 377–379, Jun. 2016.

[13] Y. Inoue, T. Sekine, and H. Asai, "Parallel-distributed block-LIM for transient simulation of tightly coupled transmission lines," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 3, no. 4, pp. 670–677, Apr. 2013.

[14] T. Takasaki, T. Sekine, and H. Asai, "Fast transient analysis of nonuniform multiconductor transmission lines using HIE-block-LIM," *IEEE Microw. Wireless Compon. Lett.*, vol. 23, no. 10, pp. 512–514, Oct. 2013.

[15] D. Klokotov and J. E. Schutt-Aine, "Latency insertion method (LIM) for electro-thermal analysis of 3-D integrated systems at pre-layout design stages," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 3, no. 7, pp. 1138–1147, Jul. 2013.

[16] J. E. Schutt-Ainé, P. Liu, J. Tan, and A. Varma, "IBIS simulation using the latency insertion method (LIM)," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 3, no. 7, pp. 1228–1236, Jul. 2013.

[17] T. M. Comberiate and J. E. Schutt-Ainé, "LIM2X: Generating X-parameters in the time domain using the latency insertion method," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 4, no. 7, pp. 1136–1143, Jul. 2014.

[18] P. Goh and J. E. Schutt-Ainé, "The latency insertion method for simulations of phase-locked loops," *J. Comput. Electron.*, vol. 13, no. 2, pp. 529–536, Jun. 2014.

[19] A. Benigni, A. Monti, and R. A. Dougal, "Latency-based approach to the simulation of large power electronics systems," *IEEE Trans. Power Electron.*, vol. 29, no. 6, pp. 3201–3213, Jun. 2014.

[20] M. Milton and A. Benigni, "Latency insertion method based real-time simulation of power electronic systems," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 7166–7177, Aug. 2018.
[21] A. F. Paterson and T. D. Anthopoulos, "Enabling thin-film transistor technologies and the device metrics that matter," *Nature Commun.*, vol. 9, no. 1, pp. 1–4, Dec. 2018.
[22] X. Guo, Y. Xu, S. Ogier, T. N. Ng, M. Caironi, A. Perinot, L. Li, J. Zhao, W. Tang, R. A. Sporea, A. Nejim, J. Carrabina, P. Cain, and F. Yan, "Current status and opportunities of organic thin-film transistor technologies," *IEEE Trans. Electron Devices*, vol. 64, no. 5, pp. 1906–1921, May 2017.
[23] E. Fortunato, P. Barquinha, and R. Martins, "Oxide semiconductor thin-film transistors: A review of recent advances," *Adv. Mater.*, vol. 24, no. 22, pp. 2945–2986, Jun. 2012.
[24] S. N. Lalgudi and M. Swaminathan, "Analytical stability condition of the latency insertion method for nonuniform GLC circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 9, pp. 937–941, Sep. 2008.
[25] W. C. Chin, A. Pashkovich, K. Malinauskas, J. E. Schutt-Aine, H. Ibrahim, N. S. Ahmad, and P. Goh, "Voltage-in-current latency insertion method for diodes and MOSFETs," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 10, pp. 1708–1720, Oct. 2020.
[26] *SPICE Models Manual*, Silvaco, Santa Clara, CA, USA, 2019.
[27] J. Schutt-Aine and P. Goh, "LIM algorithms for diodes and branch capacitors," in *Proc. IEEE Electr. Design Adv. Packag. Syst. Symp. (EDAPS)*, Dec. 2018, pp. 1–3.

WEI CHUN CHIN received the B.Eng. degree in electronic engineering from Universiti Sains Malaysia, Engineering Campus, Nibong Tebal, Pulau Pinang, Malaysia, in 2018, where he is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering.

He specializes his research in the signal and power integrity of high-speed digital designs. He is putting efforts towards the implementation of better simulation algorithms for high-speed and large networks. He was a recipient of the Gold Medal Award for B.Eng. degree in electronic engineering at Universiti Sains Malaysia, in 2018.

ANDREI PASHKOVICH (Member, IEEE) graduated from the Radiophysics and Electronics Department, Belarusian State University, Belarus, in 1993.

From 1993 to 1999, he was a Staff Scientist and worked on various aspects of machine learning algorithms, image recognition, and optical measuring systems. From 2000 to 2014, he was a SmartSpice Circuit Simulator Developer. From 2014 to 2019, he was the Director of the Circuit Simulation Division, Silvaco Inc. He is currently a fellow at Silvaco and specializes in the development of advanced circuit simulation algorithms. He specializes in circuit simulation algorithms, linear solvers, and semiconductor device models.

JOSÉ E. SCHUTT-AINÉ (Fellow, IEEE) received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1981, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 1984 and 1988, respectively.

From 1981 to 1983, he was with the Hewlett Packard Technology Center, Santa Rosa, CA, USA, as an Application Engineer, where he was involved in research on microwave transistors and high-frequency circuits. In 1988, he joined the Electrical and Computer Engineering Department, UIUC, as a member of the Electromagnetics and Coordinated Science Laboratories, where he is currently involved in research on signal integrity for high-speed digital and high-frequency applications. He is a consultant for several corporations. His current research interests include the study of signal integrity and the generation of computer-aided design tools for high-speed digital systems.

Dr. Schutt-Ainé was a recipient of several research awards, including the 1991 National Science Foundation MRI Award, the National Aeronautics and Space Administration Faculty Award for Research, in 1992, the NSF MCAA Award, in 1996, and the UIUC-National Center for Superconducting Applications Faculty Fellow Award, in 2000.

NUR SYAZREEN AHMAD (Member, IEEE) received the B.Eng. degree (Hons.) in electrical and electronic engineering and the Ph.D. degree in control systems from The University of Manchester, U.K., in 2009 and 2012, respectively.

She is currently with the School of Electrical and Electronic Engineering, University Sains Malaysia. Her current research interests include robust constrained control, intelligent control systems, computer-based control, and autonomous mobile systems in wireless sensor networks.

PATRICK GOH received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2007, 2009, and 2012, respectively.

Since 2012, he has been with the School of Electrical and Electronic Engineering, Universiti Sains Malaysia, where he currently specializes in the study of signal integrity for high-speed digital designs. His research interest includes the development of circuit simulation algorithms for computer-aided design tools. He was a recipient of the Harold L. Olesen Award, in 2010, and the Raj Mittra Award, in 2012. He has served on the technical program committee and international program committee for various IEEE and non-IEEE conferences around the world.