# An Efficient Reliability Simulation Flow for Evaluating the Hot Carrier Injection Effect in CMOS VLSI Circuits

Mehdi Kamal[1], Qing Xie[2], Massoud Pedram[2], Ali Afzali-Kusha[1], Saeed Safari[1]
[1] School of Electrical and Computer Engineering, University of Tehran, Iran
[2] Department of Electrical Engineering - Systems, University of Southern California, USA
{mehdikamal, afzali, saeed}@ut.ac.ir, {xqing, pedram}@usc.edu

***Abstract*— Hot carrier injection (HCI) effect is one of the major reliability concerns in VLSI circuits. This paper presents a scalable reliability simulation flow, including a logic cell characterization method and an efficient full chip simulation method, to analyze the HCI-induced transistor aging with a fast run time and high accuracy. The transistor-level HCI effect is modeled based on the Reaction-Diffusion (R-D) framework. The gate-level HCI impact characterization method combines HSpice simulation and piecewise linear curve fitting. The proposed characterization method reveals that the HCI effect on some transistors is much more significant than the others according to the logic cell structure. Additionally, during the circuit simulation, pertinent transitions are identified and all cells in the circuit are classified into two groups: critical and non-critical. The proposed method reduces the simulation time while maintaining high accuracy by applying fine granularity simulation time steps to the critical cells and coarse granularity ones to the non-critical cells in the circuit.**

## I. INTRODUCTION

Circuit reliability is one of the major concerns in VLSI circuits and systems designs. Hot carrier injection (HCI) effect, which causes MOSFET aging due to the deleterious effect on threshold voltage and the driving current, becomes one of the major concerns on the reliability degradation [1]. An accurate and efficient simulator with the HCI effect modeling methodology is therefore crucial for predicting the circuit performance and reliability and hinting the circuit design and manufacturing.

Extensive studies have been conducted during the past decades to model the physical phenomenon of HCI [2][3][4]. The reaction-diffusion (R-D) framework has provided theoretical understanding of the HCI effect. In this model, the rate of hot carrier generation increases as $t^n$ where $n$ is approximately 0.5 [4]. In classic Lucky Electron Model (LEM) [3], the HCI effect is characterized based on subthreshold current ($I_{sub}$). However, in modern CMOS technologies, the $I_{sub}$-based method is not appropriate since other types of leakage current such as gate tunneling leakage, source/drain junction leakage, and gate induced drain leakage become important [4]. To overcome this issue, we characterize the impact of HCI, *i.e.*, aging of VLSI circuits, using the drift of the threshold voltage and degradation of carrier mobility. The R-D framework provides efficient and accurate estimation of the impact of HCI [4].

Simulation tools for reliability of VLSI circuits considering the HCI effect have been developed and widely utilized in the industry [5][6]. These studies can produce accurate results for circuit performance degradation due to the HCI effect though proper calibration of process technology and SPICE-like simulation. However, in spite of being accurate, these simulation tools are computationally intensive, which prevents them from being scalable to handle today's circuit design with millions of transistors on a chip.

Some other studies about HCI aging analysis rely on the gate-level characterization [7][8]. In these works, the HCI effect is characterized and stored in look-up tables (LUTs) with entries of input slew rate, load capacitance, and input transitions. These works generally provide accurate and efficient HCI aging analysis but have the following drawbacks: 1) The HCI effect is characterized by using the saturation current of MOSFETs, which is easy to monitor but suffers from additional channel length, supply voltage and threshold voltage dependencies; 2) None of the prior work has considered the impact of $V_{th}$ and mobility ($\mu$) drifts on the terminal voltage transitions of MOSFETs although the changes in the transistor terminal voltages affects the generation rate of the interface traps during the lifetime of the transistor.

In this paper, we present a reliability simulation flow with fast run time and high accuracy. To characterize the HCI effect of a transistor, we propose an efficient HCI modeling method based on piecewise linear curving fitting and HSpice simulation. At the gate level, we propose two pruning techniques to reduce the effort of standard cell library characterization, based on the HCI effect dependency on the input slew rate, load capacitance, rise/fall/steady type of input transitions, and total transition counts. Then we introduce the concept of pertinent transition, which is the combination of input transition that does causes HCI-induced aging in a logic cell. Additionally, at the full chip level, to further improve the run time, we classify all logic cells in the benchmark circuits into two categories: critical and non-critical. We apply different time step granularity to save the simulation workload as well as maintain the accuracy.

The remainder of this paper is organized as follow. In Sections II and III we describe the proposed gate-level, and circuit-level HCI modeling. The results are discussed in Section IV. Finally, Section V concludes the paper.

## II. GATE-LEVEL MODELING

### A. Gate-level HCI Modeling

In this paper, the HCI impact is modeled based on the R-D framework, in which the process of generating the interface traps contains two main phases: *reaction* and *diffusion* [2][4][9]. Equation (1-a) and (1-b) capture the rate of the generating interface traps ($N_{it}$) and the diffusion equation

of hydrogen atoms, respectively. In these equations, $N_0$ denotes the number of the Si-H bounds in the substrate/gate oxide interface and $N_H$ is the density of the. $k_f$ and $k_r$ are the bond-breaking and bond-annealing constants, which determine the forward and backward reaction rates. Finally, $D_H$ is the diffusion constant of hydrogen atoms.

$$\frac{dN_{it}}{dt} = k_f(N_0 - N_{it}) - k_r N_H N_{it}, \quad (1\text{-a})$$

$$\frac{dN_H}{dt} = D_H(\frac{d^2N_H}{dx^2} + \frac{d^2N_H}{dy^2}), \quad (1\text{-b})$$

The consequence of the interface trap generation for MOSFETs is that the threshold voltage increases and the mobility decreases, both of which result in longer switching delay and lower performance. The relations are given by,

$$\Delta V_{th} = \frac{qN_{it}}{C_{ox}}, \quad (2\text{-a})$$

$$\mu = \frac{\mu_{eff}}{(1 + \alpha N_{it})^m}, \quad (2\text{-b})$$

In Equation 1, $k_f$ corresponds to the forward reaction rate , which has a linear relation with hole density ($P$) and an exponential relation with gate electric filed ( $E_{ox}$ )[2][4][9]. Additionally, in the case of the HCI, it also depends on the electric field ($E_m$) at the drain [4]. Therefore, $k_f$ is time-dependent and defined by,

$$k_f(t) = k_{f,0} \cdot P(t) \cdot \exp(E_{ox}(t)) \cdot \exp(-\frac{\Phi_{it}}{q\lambda_e E_m(t)}), \quad (3)$$

where $k_{f,0}$ is the model calibration parameter, $\Phi_{it}$ is the critical energy for electrons to create an interface trap, and $\lambda_e$ denotes the hot-electron mean-free path. Therefore, in AC modeling, $k_f$ changes during the modeling time due to changes of the *terminal voltages* of the MOSFET.

For each CMOS logic cell, the terminal voltages of each MOSFET depend the input slew rate (*ISR*), output capacitance ($C_{Load}$), and also, the behaviors of the other MOSFETs in the logic cell. One can use HSpice to accurately model the voltage difference of the terminals in the CMOS logic cell during each input transition. It may seem acceptable to use HSpice to do a one-time simulation, find the terminal voltage profiles, derive the increment of $N_{it}$, and reflect this increment to $V_{th}$ drift and mobility degradation. However, due to the generation of interface traps, the characteristics of the MOSFET itself (*i.e.*, $V_{th}$ and $\mu$) change. Thus one should re-run HSpice simulation and update $V_{th}$ and $\mu$ for each MOSFET alternately during the whole simulation process. For a reliability simulation of years of circuit operation, accurate simulation of the HCI effect on logic cell behavior by using HSpice takes too much time.

To overcome this issue, we propose a gate-level lifetime simulation method, which can model the HCI effect for the CMOS logic cells in an acceptable simulation time with a high accuracy. Since the proposed modeling method is at the gate-level, *ISR* of the target MOSFET as well as the $C_{Load}$ of the logic cell are considered to be fixed during the modeling process. The proposed modeling method combines piecewise linear curve fitting and HSpice simulation.

The details about gate-level HCI modeling is shown in Figure 1. We break the lifetime simulation of a CMOS logic cell into a series of variable-length time intervals and simulate a limited number of input transitions at the beginning of each time intervals using HSpice. Based on these simulations, we profile the voltage transitions on each terminal of the targeted MOSFET during the transition. Then we calculate the generation of interface traps ( $\Delta N_{it}{}'$ ) for *TransitionCount* number (which is typically 1e5) of transitions based on Equaiton 1 using the *HCIModeling* function in line 3. After that we apply a linear curve fitting technique within this time interval to calculate the generation of interface traps over the rest of the time interval. At the end of time interval, we reflect the generation of interface traps to $V_{th}$ and $\mu$ of MOSFET and start next time interval. The duration of the time interval is set adaptively depending on the rate of interface traps generation.

```
1:  Do
2:      Signals = HSpiceSimulation(Vth, μ)
3:      ΔNit' = HCIModeling(Signals, Nit , TransitionCount)
4:      ΔNit = Interpolation(ΔNit', SimulationStepTime)
5:      Nit += ΔNit
6:      Update(Vth), Update(μ)
7:      SimulationTime += SimulationStepTime + TransitionCount *
    SignalPeriod
8:      Update(SimulationStepTime)
9:  Until (SimulationTime > Max_ SimulationTime)
```

Figure 1 Pseudo code of the proposed HCI modeling method.

The *HCIModeling* function returns accurate generation of interface traps $\Delta N_{it}'$after *TransitionCount* transitions. We apply a linear fit to obtain the relation between $\Delta N_{it}{}'$ and *TransitionCount* and extrapolate to find the total $\Delta N_{it}$ generated in the whole time interval (*SimulationStepTime*). Next, $N_{it}$ , $V_{th}$ and $\mu$ are updated (lines 5, and 6), and *SimulationTime* is increased. Finally, we update the *SimulationStepTime* based on $\Delta N_{it}$. Because $N_{it}$ has a power law relation to the transition counts [2][4], we apply fine granularity, *i.e.*, the value of the *SimulationStepTime* is small (e.g., about 1 second) at the beginning of the lifetime simulation and coarse granularity later on. However, we also define an upper bound for the *SimulationStepTime* value (e.g., one day). This limitation is used to control the error of the method. We repeat line 2 to line 8 until the *Max_SimulationTime* is reached.

### *B. Standard Cell Library Characterization*

The HCI modeling method described in Section III-A is for single transistor and not scalable since the cycle by cycle HSpice simulation of the circuits with thousands of gates takes too much time, even if we apply piece-wise linear curve fitting. Therefore, we characterize the impact of HCI effect for the standard cell library using curve fitting and look-up-tables so that we can calculate the HCI effect directly in the circuit level lifetime simulation. Note that the HCI impact is different from one transistor to another. Hence, for every transistor of each library cell, we propose to use the following functions to determine the threshold voltage drift and mobility degradation,

$$\Delta V_{th,i} = F_{case}(ISR, C_{Load}, T) \quad (4\text{-a})$$

$$\Delta \mu_i = G_{case}(ISR, C_{Load}, T) \quad (4\text{-b})$$

Function $F_{i,case}$ is used to calculate the $\Delta V_{th}$ drift of the $i^{th}$ transistor of the logic cell based on the input slew rate, output capacitance, and the transition counts (*T)*, and finally, input transition type (*case)*. Similarly, function $G_{case}$ is used to

calculate the $\Delta\mu$ of the $i^{th}$ transistor. Since the HCI effect is dependent on the voltage waveforms at the transistor terminals during a transition, different input slew rate and load capacitance could lead to different interface trap generation rates and in turn different degrees of threshold voltage drift and mobility degradation. Therefore, we pick the typical input slew rate (from 10ps to 350ps) and load capacitance (0.25fF to 8fF), perform the proposed gate-level lifetime HCI modeling and record the behaviors of the threshold voltage and mobility.

Doing HCI modeling for each member of the *case* is a time consuming process. Therefore, we propose two approximations to prune the gate characterization process. First, for the parallel transistors (e.g., NOR cell) of which the drain and source terminals are connected together (e.g., Figure 2.b), we can characterize one transistor and use the characterized functions for all the other ones, under the condition that they are all identically sized. The second approximation is for the transistors that are series-connected. We observe that the voltage difference between the drain and source of the transistors that are in the upper place in the stack (e.g., Figure 2.c, transistor M1) is much greater than that of the transistors in a lower place (e.g., Figure 2.c, transistor M2). This observation originates from fact that in nanometer technology the difference between the source voltage and the threshold voltage of the transistors is becoming small (e.g., in 45 nm the source voltage is 1.1V while $V_{th}$ is about 0.47V for the NMOS). This causes a very small electrical field in the drain side of transistors that are placed in a lower place in the transistor stack, and hence, the HCI become very small for these transistors. These two pruning techniques allow us to simplify the HCI characterization process.

The output information of the logic cells is also needed in order to simulate the HCI effect at the full circuit-level. Therefore, we characterize the delay and output slew rate (OSR) for logic cells in the standard cell library by using linear curve fitting and look-up tables. Similar to the HCI effect characterization, we identify input transition type that can cause an output transition. For each case, we pick the typical input slew rate (from 10ps to 350ps) and load capacitance (0.25fF to 8fF) and perform HSpice simulation with respect to different $V_{th}$ and $\mu$. At each combination of ISR and $C_{Load}$, we measure the output slew rate as well as the delay, and apply a curve fitting to relate them to $V_{th}$ and $\mu$.

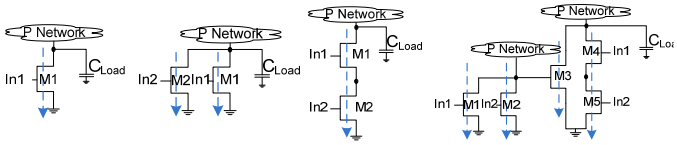
## III. Circuit-Level Modeling

The HCI impact on each logic cell inside a digital circuit depends on its transition count, the type of transitions it undergoes, slew rate of its input signals, and its $C_{Load}$. Note that $C_{Load}$ is a constant during the modeling. The input slew rate of a logic cell, which is the output slew rate of its parent cells, depends on $V_{th}$ and $\mu$ of the transistors of the parent cells. Hence, the input slew rates of a cell may change over time, which in turn affects the HCI impact on the $V_{th}$ and $\mu$. In this work, we break the lifetime simulation of the circuit into a number of predefined time slots, and during each time slots the HCI impact is calculated, as well as the input slew rate of the next-stage cell, based on current $V_{th}$ and mobility.

Since the circuits are typically very large, we propose some techniques to reduce the workload of simulation. At first we detect which transitions are pertinent. Then we analyze the logic cell structure and prune the transistors and pertinent transitions that have negligible HCI impact. Next, logic cells in the circuit are classified to decrease the runtime of the simulation. Finally, based on the pertinent transitions and cell classification, the HCI simulation is done for the circuit.

### A. Pertinent Transition Identification

As mentioned in Section III, the terminal voltage transitions have a big influence on the HCI effect. Among all types of transition, some of them are able to generate the interface traps while the others do not. Therefore, we define the types of transition that have an HCI impact as the *pertinent transitions*, and we only calculate HCI impact for these types of transitions during HCI modeling. The definition of pertinent transitions varies from one logic cell to another. Without loss of generality, here we only define it for four types of basic cells in the standard library: INV, NAND, NOR and XOR. This concept can be easily applied for any other type of logic cell. We adopt the *single input switching* (SIS) assumption, *i.e.*, the output transition for a logic cell, if it occurs, is triggered by a single input signal transition.




(a) (b) (c) (d)

Figure 2 a)INV b)NOR c)NAND d) XOR cell

For the inverter, the pertinent input transition is the one in which the input signal rises from zero to one. For each 2-input cell, we consider all eight possible transitions. Since we only focus on aging of NMOS transistor in this paper, among these eight transitions, all transitions that cause the output value to fall from one to zero are pertinent (they are the only ones that cause current flow in the NMOS transistors). In addition, some pertinent transitions result in the HCI effect even though the output does not change. This is due to current flow through some transistors to charge or discharge the internal capacitances of the logic cell. For example, the transition $(\uparrow \mathrm{In1}, \overline{\mathrm{In2}})$ affects transistor M1 in NAND and M4 in XOR. The pertinent transitions for the basic cells in standard library are listed in TABLE I. The upwards arrows denote rising transition while the downwards arrow denote falling transition.

TABLE I Pertinent States in Different Cell Types

| | |
|---|---|
| INV | $(\uparrow \mathrm{In1})$ |
| NOR | $(\uparrow \mathrm{In1}, \overline{\mathrm{In2}}), (\overline{\mathrm{In1}}, \uparrow \mathrm{In2})$ |
| NAND | $(\uparrow \mathrm{In1}, \mathrm{In2}), (\uparrow \mathrm{In1}, \overline{\mathrm{In2}}), (\mathrm{In1}, \uparrow \mathrm{In2})$ |
| XOR | $(\uparrow \mathrm{In1}, \mathrm{In2}), (\uparrow \mathrm{In1}, \overline{\mathrm{In2}}), (\mathrm{In1}, \uparrow \mathrm{In2}), (\overline{\mathrm{In1}}, \uparrow \mathrm{In2}), (\downarrow \mathrm{In1}, \overline{\mathrm{In2}}), (\overline{\mathrm{In1}}, \downarrow \mathrm{In2})$ |

We simulate the pertinent states that are listed in TABLE I and investigate the magnitude of the HCI impact of each case. We find that the HCI impact on different NMOS transistors for the same pertinent transition are very different from each other due to the MOSFET structure of the pull down network of the logic cell. For example, in a transistor stack, the transistor that is near $C_{Load}$ suffers the most significant HCI impact while this impact is greatly reduced for the transistor that is closest to the

power/ground rail. Hence, among the pertinent transitions, some of them have a very small impact on the $V_{th}$ and mobility drift and, without losing the accuracy, we can ignore them. Therefore, we propose to only consider input transitions for the transistor that is near the output. In addition, we find that the HCI effect caused by charging/discharging the parasitic internal capacitances but without an output change is also negligible. This is because these capacitances are very small in the nano-scale design technology. Hence, we reduce TABLE I to TABLE II after we ignore the two types of transitions aforementioned. Note that for XOR cell, the transitions $(\uparrow \mathrm{In1}, \overline{\mathrm{In2}})$ and $(\overline{\mathrm{In1}}, \uparrow \mathrm{In2})$ generate interface traps in the M1 and M2 transistors, respectively. The main source of the current that passes through these transistors is the short circuit current, not the discharging current of the internal capacitance of the gate.

TABLE II PERTINENT STATES IN DIFFERENT CELL TYPES AFTER PRUNING THE UNIMPORTANT TRANSITIONS TYPE

| | |
|---|---|
| INV | $(\uparrow \mathrm{In1})$ |
| NOR | $(\uparrow \mathrm{In1}, \overline{\mathrm{In2}}), (\overline{\mathrm{In1}}, \uparrow \mathrm{In2})$ |
| NAND | $(\uparrow \mathrm{In1}, \mathrm{In2})$ |
| XOR | $(\uparrow \mathrm{In1}, \mathrm{In2}), (\uparrow \mathrm{In1}, \overline{\mathrm{In2}}), (\overline{\mathrm{In1}}, \uparrow \mathrm{In2}), (\downarrow \mathrm{In1}, \overline{\mathrm{In2}}), (\overline{\mathrm{In1}}, \downarrow \mathrm{In2})$ |

### B. Cell Classification

In the circuit-level simulation, we are particularly interested in the output slew rate since it will partially determine the HCI impact of the cells in the subsequent stages of the circuit. Based on the technology as well as our experiment, we find that during the gate characterization, when the $V_{th}$ drift is below 4%, the drift of output slew rate of the logic cell is small so that we can simulate the HCI impact with large time steps. Therefore, we propose a *HCI criticality classification* procedure to classify the logic cells in circuit into two groups: critical and non-critical. The HCI effect for the critical (non-critical) cells group is modeled using small (large) time steps.

The $C_{load}$ has a positive relation while the ISR has a negative relation with the $V_{th}$ drift. Thus, we define the classification criterion *ISRCR* as the ratio of ISR and $C_{load}$, which is given by,

$$ISRCR = ISR/C_{Load}.$$

We set a threshold value (denoted by $ISRCR_{th}$) for *ISRCR* to determine whether a cell is HCI-critical or not. As we will show in Section IV, the impact of the HCI is decreased with the increase of this ratio. Therefore, if the *ISRCR* value of a cell is smaller than this predefined threshold value, then this cell is critical, otherwise it is non-critical. Note that the $ISRCR_{th}$ value is obtained when the $V_{th}$ drift is 4%, thus the $ISRCR_{th}$ are different for each cell type.

### C. Circuit-level Modeling

Figure 3.a depicts the pseudo code of the proposed circuit-level HCI modeling. In this code, ΔT is the modeling time steps and α is the ratio between the modeling time steps of the critical to non-critical cells in the circuit. During the HCI modeling (Figure 3.a lines 7 and 10), since the library characterizing has been done for some discrete values of the ISR and $C_{Load}$, we do a linear-interpolation to extract the $V_{th}$ and μ drifts if the ISR and $C_{Load}$ fall in between the entries of the LUTs. Assume the library characterization is done based on $SC_{Load}$ and SISR sets:

$$SC_{Load} = \{C_{Load,1}, \dots, C_{Load,N}\}, \qquad SISR = \{ISR_1, \dots, ISR_M\},$$

The $V_{th}$ and μ drifts of $i^{th}$ cell, with output capacitance of $C_{Load,i}$ and input slew rate of the $j^{th}$ input as $ISR_{i,j}$, is bounded by nearest neighbors that are defined by

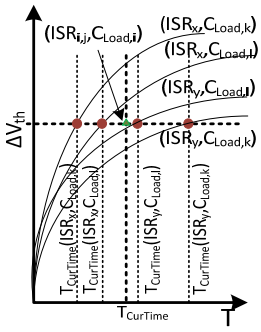
$$C_{Load,k} \le C_{Load,i} \le C_{Load,l} \ \ where\ 1 \le k, l \le N$$

$$ISR_x \le ISR_{i,j} \le ISR_y \ \ where\ 1 \le x, y \le M$$

The four nearest neighbors (*i.e.*, $\{(ISR_x, C_{Load,k}), (ISR_x, C_{Load,l}), (ISR_y, C_{Load,k}), (ISR_y, C_{Load,l})\}$) have distinct $V_{th}$ drift and μ degradation relation with respect to the transition counts. More precisely, for the same number of transitions counts, the level of $\Delta V_{th}$ and Δμ are different for each case (shown as the vertical dashed line in Figure 3.b). However, it is not appropriate to simply take a weight average using the crossing points of four trend lines and the vertical dashed line because the $\Delta V_{th}$ and Δμ depend on the current aging level of $V_{th}$ and μ. To solve this issue, we determine the equivalent transition counts of these four curves by searching the points that have the same aging effect for the $i^{th}$ cell (shown as red points in Figure 3.b) and calculate the $\Delta V_{th}$ and Δμ of these four nearest neighbor using Equation 4. Then we take a weight average of these four values to find the $V_{th}$ and μ drifts.

```
1:Estimate ISR of the logic cells
2:Grouping the logic cells based on ISRCR_th
3:Do
4:   Simulating input vectors
5:   Obtain pertinent transitions
6:   If (CurTime = TimeofAllCell )
7:      Do HCI modeling for all cells
8:      TimeofAllCell += αΔT
9:   Else
10:        Do HCI modeling for critical cells
11:  End if
12:  Update the input slew rates
13:  CurTime += ΔT
14:Until( CurTime > MaxTime)
```

(a)




(b)

Figure 3 a) Pseudo code of the circuit-level HCI modeling b) $T_{CurTime}$ under ISR and $C_{Load}$ values of the nearest neighbors.

## IV. EXPERIMENTAL RESULTS

To evaluate the proposed gate-level HCI characterization approach, we study the impacts of the HCI on the NMOS transistors in standard 45nm CMOS cells library [10] under different ISR values and $C_{Load}$ values. We do the simulation for 4e16 transitions, which is equivalent to 10 years simulation in the typical operating conditions, where the circuit is running all the time at 1 GHz frequency with an activity factor of 0.05 to 0.1. In gate-level characterization, we apply adaptive time step duration, starting at 1s and increasing gradually (Figure 1 Line 8) since the HCI degradation becomes slower. The duration of the time step can be used to trade off the characterization accuracy and runtime. Figure 4 shows the accuracy of the HCI modeling of the NMOS transistor of the Inverter cell at different durations of simulation time step. The results show that the maximum error appears at the beginning since the HCI degrades faster at that moment. We define one day as the upper

bound of the duration of the simulation time step since the errors of the durations larger than one day are much higher.

As we mentioned before, due to the change of $V_{th}$ and mobility, the $V_{ds}$ transition also changes, which in turn affects the HCI impact during the lifetime of the transistor. This fact requires us re-simulate the terminal voltages of the transistor using HSpice during the HCI modeling. Figure 5 shows the difference of two modeling approaches: considering change of $V_{ds}$ transition and not considering change of $V_{ds}$ transition. The results show that the error increases as the output capacitance increases. When the input slew rate increases, the error exhibits two different trends. For the output capacitance that is greater than 4 fF, by increasing the ISR, the error increases. Otherwise, the error decreases. Therefore, this result shows the importance of considering the change of $V_{ds}$ and the dependence of $V_{ds}$ change on both input slew rate and output capacitance.

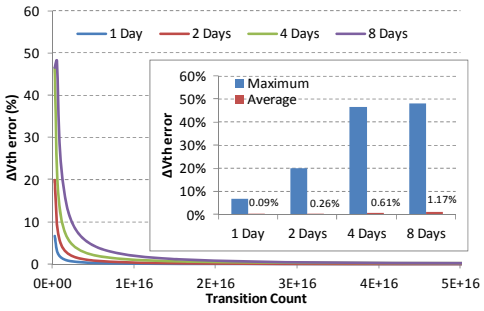



Figure 4 $\Delta V_{th}$ error under diffent maximum time step interval compared to the case when the maximum time step duration is half of a day.

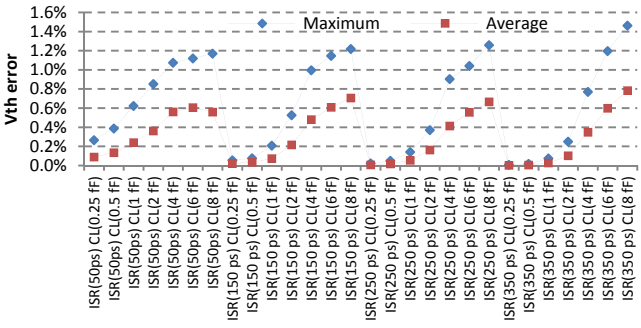



Figure 5 The maximum and average of the $\Delta V_{th}$ error when the $V_{ds}$ behavoir is assumed as a constant value.CL stands of $C_{Load}$.

Figure 6shows the $V_{th}$ and μ drifts of the NMOS transistors in standard cells library due to the pertinent transitions. The results show that generally as the output capacitance increases, the HCI effect on $V_{th}$ drift and μ degradation increase. In contrast, by increasing the input slew rate (*i.e.*, slow input), the HCI has less impact on $V_{th}$ drift and μ degradation. Note that the $\Delta V_{th}$ and $\Delta\mu$ of the NMOS transistors of NOR cells are similar to INV and are not listed here. The impact of the HCI on the $V_{th}$ and mobility drift for NAND cell is smaller, compared to the INV cells. This result originates from the fact that the $V_{ds}$ value of the NMOS in the NAND cell due to the stacking is smaller than those of the INV and NOR cells during the discharging the $C_{Load}$.

TABLE III shows that one pertinent transition can result in different HCI impacts on transistors for the NAND cell due to the structure. Compared to M1 (Figure 2.c), the impact of the HCI on the M2 is negligible since the electrical field in the drain side of M2 is much smaller than that of the M1.

TABLE III THE $\Delta V_{TH}$ VALUE FOR THE DIFFERENT TRANSISTORS OF THE NAND CELLS UNDER TWO INPERTINENT TRANSITIONS

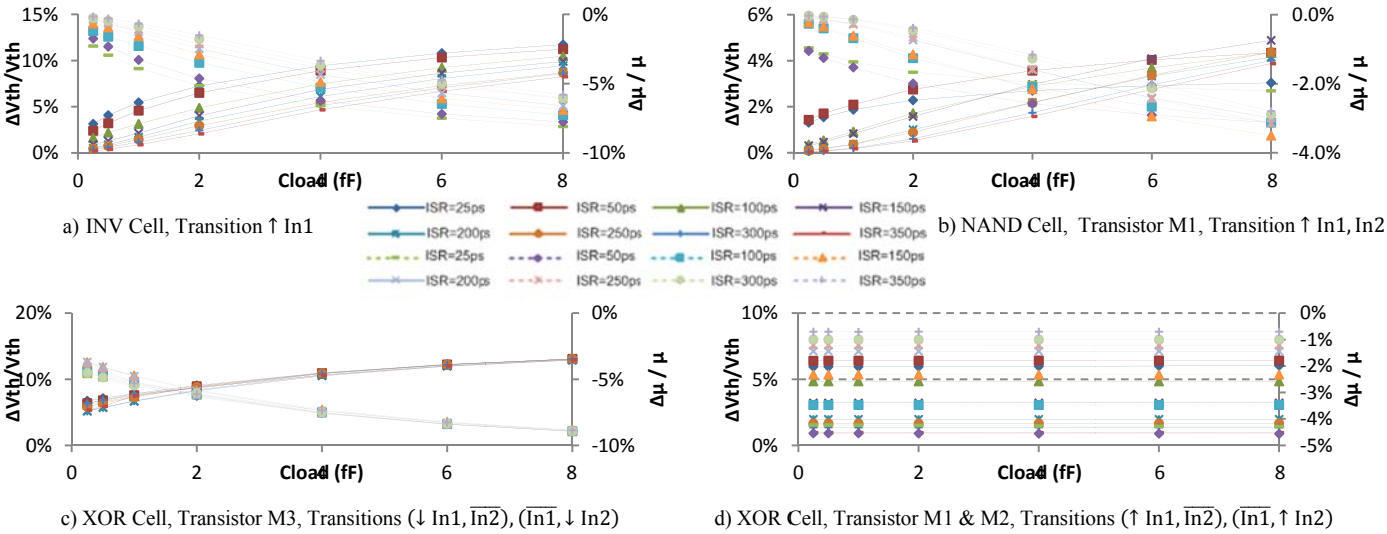
| | Transition Case | M1 | M2 |
|---|---|---|---|
| NAND | ↑In1, In2 = 0 | 9.40E-04 | 0 |
| | In1 = 1, ↑In2 | 1.25E-04 | 3.94E-12 |

For XOR cell, the impact of the transitions on the M4 and M5 transistors are similar to the case of the NAND cell. The pertinent transitions ($\downarrow \mathrm{In1}, \overline{\mathrm{In2}}$) and ($\overline{\mathrm{In1}}, \downarrow \mathrm{In2}$) have an HCI impact on the M3 transistor. Note that in Figure 6.c, unlike the other three gates, the HCI impact for XOR cell is independent of the input slew rate, especially at the high output capacitance. This is because that in this case, the transition triggered by the left P network of the logic cell, which leads to nearly constant terminal voltage transition. In addition, transitions ($\uparrow \mathrm{In1}, \overline{\mathrm{In2}}$) and ($\overline{\mathrm{In1}}, \uparrow \mathrm{In2}$) have HCI impact on the transistors M1 and M2, respectively. Since in this case the output capacitance does not have any effect on the terminal voltage of theses transistors, the HCI impact is independent from the output capacitance. However, the input slew rates have a direct impact on them. By increasing the input slew rate, the $V_{th}$ drift and mobility degradation increase, as shown in Figure 6.d.

Figure 7 shows the $V_{th}$ drift of the three cells under different IRSCR ratios. In this paper, the $ISRCR_{th}$ is set as the point where the percentage change of $V_{th}$ is smaller than 4%. According to our simulation, the $ISRCR_{th}$ value for the INV, NOR, and NAND cells are 100, 200, and 50, respectively. These values are used as criterions to classify whether a cell is the critical and non-critical.

To evaluate the circuit-level modeling, we select two C17 and S27 small ISCAS benchmarks and simulate the HCI effect using the proposed flow. The inputs to the circuits are 1,000 randomly generated binary signals and we simulate 4e16 transitions. The primary input slew rates are all set to 100ps. We compare the results of the proposed simulation flow to the accurate simulation results, which is obtained using HSpice simulation without any LUTs. Also, the evaluation was done for two cases: 1) not considering the cells classification (*i.e.*, it is equivalent as considering all cells as critical cells), and 2), considering the cells classification. The proposed circuit level is implemented in C++ and run on a PC with 3.2 GHz processor and 32GB RAM. The runtime and the accuracy of the circuit level modeling are reported in TABLE IV. The results show the runtime of the proposed simulation method has an acceleration of ~3e6X and accuracy of the proposed method is within ±30% (1.5 times smaller than the method proposed in [8]). The results of case 2 versus case 1 show a 2X simulation runtime reduction.

We simulate the HCI impact on some big benchmarks of the ITC99 package to show the scalability of the proposed simulation flow. We report the results in TABLE V. The inputs of the circuit were generated in the same way as C17 and S27 benchmarks, and the gates are classified based on the $ISRCR_{th}$ values. The results show the runtime of the proposed method is very fast and in the worst-case is below 1.5 minutes.




a) INV Cell, Transition ↑ In1

b) NAND Cell, Transistor M1, Transition ↑ In1, In2

c) XOR Cell, Transistor M3, Transitions (↓ In1, $\overline{In2}$), ($\overline{In1}$, ↓ In2)

d) XOR Cell, Transistor M1 & M2, Transitions (↑ In1, $\overline{In2}$), ($\overline{In1}$, ↑ In2)

Figure 6 $V_{th}$ and μ variation under different ISR and $C_{Load}$. In the charts, the solid lines belong to the $\Delta V_{th}/V_{th}$ while the dashed lines belong to the $\Delta\mu/\mu$.

TABLE IV THE RUNTIME AND THE ACCURACY OF THE PROPOSED CIRCUIT-LEVEL MODELING

| benchmark | Number of Cells | Number of Critical Cells | Modeling Runtime | | Accurate Runtime | ΔVth and Δμ error | |
|---|---|---|---|---|---|---|---|
| | | | Non- Classification | Classification | | Non-classified | Classified |
| C17 | 7 | 2 | 0.263(s) | 0.062(s) | ~4 days | -28.63% to 25.09 | -29.06% to 25.7% |
| S27 | 10 | 4 | 0.33(s) | 0.177(s) | ~7 days | -26.4% to 27.93% | -30.93% to 24.39% |

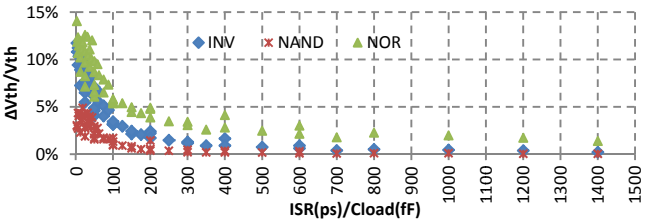



Figure 7 $V_{th}$ drift under diffent Input Slew Rate(ISR) to $C_{Load}$ ratios.

TABLE V DELAY DEGRADATION DUE TO THE HCI IMPACT AND THE RUNTIME OF THE PROPOSED METHOD

| Benchmark | Gates Count | ΔD | Run Time (s) |
|---|---|---|---|
| b13 | 363 | 0.52% | 1.26 |
| b14 | 8812 | 0.56% | 21.73 |
| b15 | 8371 | 0.58% | 25.61 |
| b17 | 29267 | 1% | 83.57 |
| b20 | 17648 | 0.83% | 46.88 |
| b21 | 17972 | 0.67% | 47.41 |

## V. CONCLUSION

As the scaling down of the transistor sizes, the impact of the HCI becomes an issue and gets renewed interests in nano-scale design. This paper suggested a scalable reliability simulation flow to analyze the HCI induced transistor aging with a fast run time and high accuracy. In the transistor-level, the reaction-diffusion HCI modeling was used. In gate-level, a characterization method based on the accurate HSpice modeling in combination with piecewise linear curve fitting was proposed to reach both accurate results and short simulation runtime. By using this method, the impact of the HCI on the four types of the logic cells (*i.e.*, INV, NAND, NOT and XOR) was studied. For each case, we also showed which kind of the transitions are important so that they should be considered in HCI modeling. Additionally, for circuit-level HCI modeling, we propose a classification method that adaptively controls the granularity of the simulation time steps. The simulation method improves the simulation runtime against the accurate modeling method by several orders of magnitudes with an accuracy of ±30%.

## REFERENCES


[1] X. Li, J. Qin, and J.B. Bernstein,"Compact Modeling of MOSFET Wearout Mechanisms for Circuit-Relaibility Simulation," in IEEE Transactions in Device and Materials Reliability, Vol(8), pp. 98-121, 2008.

[2] H. Kufluoglu and M. Ashraful Alam, "A Generalized Reaction–Diffusion Model With Explicit H-$H_2$ Dynamics for Nagative-Bias Temperature-Instability Degradation," in IEEE Tran. on Electron Devices, Vol(54), pp. 1101-1107.

[3] C. Hu, *et al.*, "Hot-Electron-Induced MOSFET Degradation – Model, Monitor, and Improvement", in IEEE Tran. on Electron Devices, Vol(32), pp. 375-385.

[4] W. Wang, *et al.*, " Comapct Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology," in IEEE Transactions on Device and Materials Reliability, Vol(7), pp. 509- 517, 2007.

[5] R. Tu, *et al.*, "Berkeley Reliability Tools - BERT", IEEE Trans. CAD IC System, Vol. 12, No. 10, pp. 1524, 1993.

[6] V. Huard, *et al.*, "CMOS Device Design-In Reliability Approach in Advanced Nodes," in Proceedings of the 47th Annual International Reliability Physics Symposium, 2009, pp. 624-633.

[7] D. Lorenz, *et al.*, "Aging analysis at gate and macro cell level" in Proceedings of ICCAD, 2010, pp. 77-84.

[8] J. Fang and S.S. Sapatnekar, "The Impact of Hot Carriers on Timing in Large Circuits," in Proceedings of ASP-DAC, 2012, pp. 591-596.

[9] H. Kufluoglu and M. Ashraful Alam, "A Geometrical Unification of theories of NBTI and HCI Time-Exponents And Its Implications for Ultra-Scaled Planar and Surround-Gate MOSFETs," in Proceedings of the IEEE International Electron Devices Meeting (IEDM), 2004, pp. 113-116.

[10] Nangate 45nm Open Cell Library. http://www.nangate.com/