SAMURAI: An accurate method for modelling and simulating non-stationary Random Telegraph Noise in SRAMs

Karthik .V. Aadithya*‡, Alper Demir†, Sriramkumar Venugopalan* and Jaijeet Roychowdhury*
*Department of Electrical Engineering and Computer Sciences, The University of California, Berkeley, CA, USA
†Department of Electrical and Electronics Engineering, Koç University, Istanbul, Turkey
‡Contact author. Email: aadithya@eecs.berkeley.edu

***Abstract*—In latest CMOS technologies, Random Telegraph Noise (RTN) has emerged as an important challenge for SRAM design. Due to rapidly shrinking device sizes and heightened variability, analytical approaches are no longer applicable for characterising the circuit-level impact of non-stationary RTN. Accordingly, this paper presents SAMURAI, a computational method for accurate, trap-level, non-stationary analysis of RTN in SRAMs. The core of SAMURAI is a technique called Markov Uniformisation, which extends stochastic simulation ideas from the biological community and applies them to generate realistic traces of non-stationary RTN in SRAM cells. To the best of our knowledge, SAMURAI is the first computational approach that employs detailed trap-level stochastic RTN generation models to obtain accurate traces of non-stationary RTN at the circuit level. We have also developed a methodology that integrates SAMURAI and SPICE to achieve a simulation-driven approach to RTN characterisation in SRAM cells under (a) arbitrary trap populations, and (b) arbitrarily time-varying bias conditions. Our implementation of this methodology demonstrates that SAMURAI is capable of accurately predicting non-stationary RTN effects such as write errors in SRAM cells.**

I. Introduction and Motivation

SRAMs[1] find application in several domains including CPU caches, LCD screens and on-chip memories for both ASICs and FPGAs. This is because SRAMs offer several advantages over their dynamic counterparts: they are much faster, they consume less power and they do not need complex circuitry to periodically refresh their states.

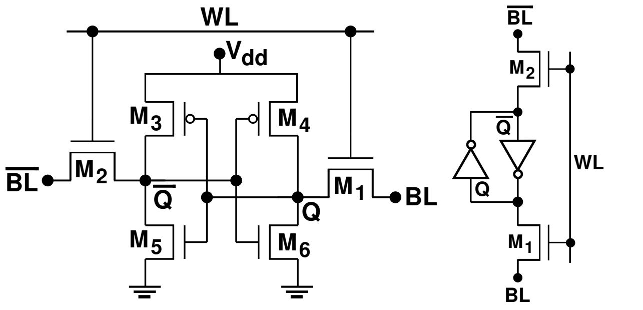

Fig. 1. A 6T SRAM cell.

Fig. 1 (left) shows the schematic for an SRAM cell that stores one bit (Q). The core of this SRAM cell is a cross-coupled pair of inverters formed by the four transistors M3-M6 (Fig. 1 right). The external circuit can read and write Q by controlling the pass transistors M1 and M2.

A. RTN as an emerging SRAM design challenge

In deeply scaled technologies, several non-idealities limit the minimum supply voltage (V_{dd}) under which an SRAM cell can be operated. Fig. 2 depicts the increase in V_{dd} necessitated by different non-idealities, as a function of CMOS technology [1]. Each CMOS technology is represented by a stacked bar, to which the effects of different non-idealities such as (a) V_T shifts due to global and local parameter variations, (b) Negative Bias Temperature Instability (NBTI), and (c) Random Telegraph Noise (RTN) are successively added (on top of the nominal supply voltage

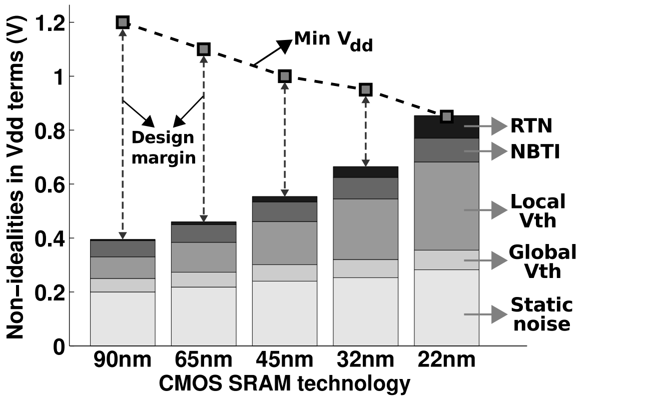

Fig. 2. Impact of non-idealities (quantified in V_{dd} terms) on SRAM design margins under different CMOS technologies. Data courtesy: Y. Tsukamoto, Renesas Electronics Corp., Japan.

that overcomes static noise). Also included is a plot (the downward sloping dashed line) depicting the scaling of V_{dd} at different technologies.

As seen from the figure, the impact (in V_{dd} terms) of RTN on SRAMs has been steadily increasing under continued CMOS scaling. Moreover, coming on top of the other non-idealities, it is the incremental contribution made by RTN that is poised to push the minimum supply voltage over the dashed line that represents V_{dd} scaling. If this happens, RTN would reduce the SRAM design margin to zero and also render any further voltage scaling impossible. Therefore, to ensure continued CMOS scaling, RTN is of enormous significance even though its magnitude is small compared to other non-idealities.

B. Towards overcoming RTN

Although RTN is a severe limiting factor (as seen from Fig. 2), the following three observations suggest strategies for coping with it:

Correlation between RTN and NBTI: Recent evidence [1] suggests that RTN and NBTI are positively correlated. As a result, the total design margin impact of RTN and NBTI, taken together, is likely to be smaller than the sum of their individual design margin impacts [1]. In Fig. 2, this corresponds to an overlap between the top two boxes of each stack, which, if accurately taken into account, would enable more design choices.

Pessimism of stationary RTN analysis (hence the need for non-stationary analysis): From measurement data, it is well-known that stationary RTN analysis harbours considerable pessimism (the difference between predicted and observed noise power is sometimes as high as 15dB) [2], [3]. Hence it is likely that non-stationary RTN analysis techniques (such as the one we propose in this paper) would open up more design choices, by virtue of being more accurate in real-world (non-stationary) operating environments.

The case for computational RTN characterisation: RTN is caused by the random capture and release of charge carriers by traps located in a MOS transistor's oxide layer [4], [5]. Analytical approaches to RTN characterisation are based on the assumption that a large number of active traps exist in the dielectric. Under this assumption, statistical averaging

[1]Static Random Access Memories

978-3-9810801-7-9/DATE11/©2011 EDAA

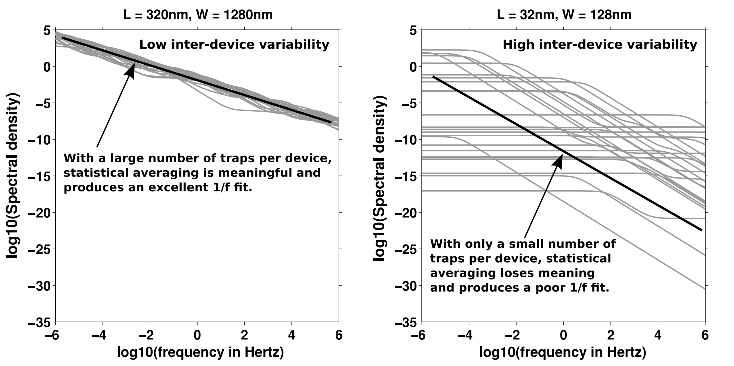

Fig. 3. Spectral density plots for 25 randomly sampled devices in two CMOS technologies.

shows that RTN obeys $1/f$ characteristics [4], [5]. However, with increasingly small device sizes, this assumption is no longer valid [6]–[8]. Indeed, detailed models for trap profiling (corroborated by measured data) suggest that in deeply scaled CMOS technologies, only about 5-10 traps are active at any given bias point [6], [7]. As a result, the analytical $1/f$ fit is often a poor model for highly scaled devices.

Fig. 3 illustrates this point using spectral density plots for 25 device instances (randomly sampled using the trap profiling model of [6], and held at constant bias) from two CMOS technologies. While the analytical solution is clearly a good fit for the older technology (left), it completely fails to capture the trap profiles of devices in the newer technology (right).

Although Fig. 3 shows that analytical approaches are no longer applicable, the encouraging observation is that this represents a clear case for computational approaches based on stochastic simulation [9] of trap activities. These approaches (such as the one proposed in this paper) are ideally suited to handle the small trap populations in today's highly scaled devices. Indeed, computational approaches actually take advantage of the small trap populations to achieve greater efficiency.

C. What does accurate RTN characterisation entail?

The design of RTN-tolerant future generation SRAMs depends critically on our ability to leverage the above three observations. For this we need:

An accurate model for RTN: It is known that both RTN and NBTI originate due to traps in the MOS oxide layer. The correlation between RTN and NBTI is most likely due to this *common root cause* [1]. Therefore, an RTN model based on first principles (*i.e.*, the capture and emission of electrons by traps in the oxide layer) is likely to succeed in accurately capturing the NBTI correlation. Whereas detailed equations describing RTN generation from first principles at the device level are already available [5], [6], this paper, to the best of our knowledge, is the first attempt at incorporating such sophisticated models into a tool for RTN characterisation at the circuit level.

A computational method for predicting the circuit-level impact of non-stationary RTN: Today, the most advanced computational approach for RTN is that of Ye et. al. [10], which works by generating RTN-like waveforms starting from ideal white-noise sources. The main advantage of this method is that it integrates RTN simulation with SPICE-level circuit simulation. However, a key drawback of this method is that it is incapable of taking into account the bias-dependent, non-stationary statistics of RTN, which play a crucial role during SRAM operation (see next subsection). Moreover, the white noise sources drastically reduce the efficiency of this method from a simulation perspective. Indeed, to date, the only reported SRAM application of this method has been to analyse the simplest case of *a single trap* in an entire SRAM cell, and that too operating under constant bias assumptions.

By contrast, we develop a technique (§III) for generating genuinely non-stationary RTN. This technique, based on uniformisation [11]–[13] of a trap-level Markov chain model (§II), provably generates RTN traces that are (stochastically) exactly identical to the RTN physically measured on fabricated SRAMs. Hence the title of this paper, SAMURAI, which stands for **S**RAM **A**nalysis by **M**arkov **U**niformisation with **R**TN **A**wareness **I**ncorporated. While being a computational method based on trap-level first principles, SAMURAI is capable of accurately simulating non-stationary RTN at the circuit level under (a) arbitrary trap populations, and (b) arbitrarily time-varying bias conditions. Moreover, we are able to integrate SAMURAI with SPICE, without encountering efficiency issues, to conduct full-fledged RTN analysis of SRAMs with varying trap populations under realistic, non-stationary operating environments.

D. Non-stationarity of RTN in SRAMs: A closer look

We have already mentioned that non-stationary analysis, being less pessimistic, often enables more design choices. We now illustrate the critical importance of non-stationary RTN, in the context of SRAMs.

In an SRAM cell, RTN can produce two adverse effects [14], [15]: (1) it can *slow down* write operations, and (2) it can cause *write errors*[2].

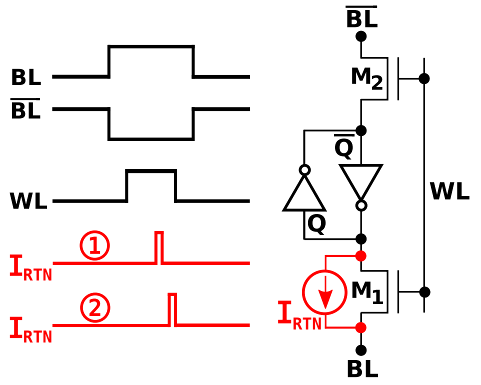

Fig. 4. RTN in pass transistor M1 modelled as a glitch I_{RTN} while a 1 is being written to the SRAM cell.

Consider the pass transistor M1, whose RTN can be modelled as a current source I_{RTN} that opposes the nominal transistor current, as illustrated in Fig. 4 (right). The top three waveforms in Fig. 4 (left) are the signals applied to write a 1 to the SRAM cell, while the bottom two waveforms (numbered 1 and 2) represent two possible I_{RTN} "glitches".

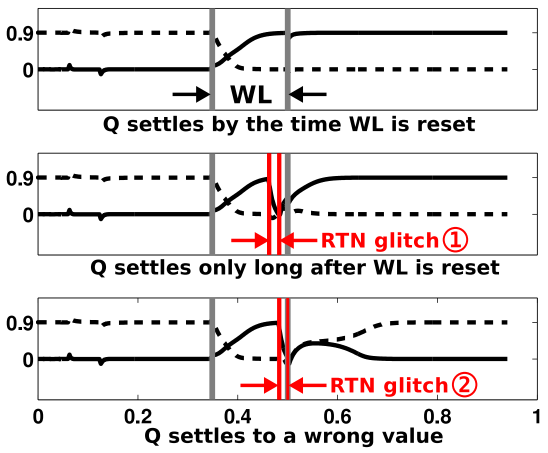

Fig. 5. RTN in the pass transistor can either (i) slow down the write operation, or (ii) result in a write error. X-axis: Time (ns). Y-axis: Voltage (V). The solid line represents Q while the dotted line represents $\overline{Q}$.

Fig. 5 shows BSIM-4 SPICE simulations for the different I_{RTN} scenarios. The top portion shows that in the absence of I_{RTN} glitches, the signals

[2]RTN-induced SRAM read failures have also been reported [16]. SAMURAI is capable of predicting these too; however, due to space constraints, we do not discuss read failures in this paper.

Q (the solid line) and $\overline{Q}$ (the dotted line) settle to their correct values by the time WL is de-asserted. The middle portion shows that if a glitch starts *after* WL is asserted, but ends *before* WL is de-asserted, it can *slow down* the write operation, *i.e.*, Q does not assume its correct value until long after WL is reset (hence a read operation initiated in the interim can upset the stored value). The bottom portion shows that if a glitch starts *just before* WL is de-asserted, and continues until WL is de-asserted, it can result in a *write error*.

From the above discussion, it is clear that the *timing of RTN glitches* plays a crucial role in deciding whether an SRAM cell is compromised or not. In other words, there are certain *critical moments* (*e.g.*, during switching) when an SRAM cell is extremely sensitive to RTN spikes [15]. At other times, even a reasonably large RTN spike would not produce any observable effect. Thus, the key to successful SRAM design lies in understanding the RTN patterns during such *critical moments*.

Moreover, during such *critical moments* in an SRAM cell's duty cycle, all 6 transistors experience large and rapid bias swings. Under such fast bias variations, the traditional stationarity assumptions that simplify RTN analysis are no longer valid. Therefore, especially in the SRAM context, a non-stationary RTN analysis technique (such as SAMURAI) is the need of the hour. In §IV, we show that SAMURAI can indeed predict non-stationary effects such as SRAM write errors.

E. SAMURAI: Structure, capabilities and summary of results

In previous subsections, we highlighted the need for an accurate, trap-level, non-stationary, computational RTN characterisation technique, with application to SRAM design. Against this background, in this paper, we develop SAMURAI, a computational tool for accurate modelling and simulation of non-stationary RTN. Starting from first principles (the capture/release of electrons by MOS oxide traps), SAMURAI generates accurate RTN traces for entire circuits (such as SRAMs) under arbitrary trap populations and arbitrarily time-varying bias conditions.

SAMURAI is a computational approach based on Markov Uniformisation [11]–[13], which is an extension of Gillespie's stochastic simulation algorithm [9] to handle non-stationarity. Such computational approaches are well-known in the biological community, where they are used for accurate simulation of biochemical reactions involving a small number of molecules [9], [17], [18]. Analogously, we use SAMURAI for accurate simulation of RTN in SRAMs involving a small number of traps. Specifically, we have developed a simulation-driven methodology (illustrated in Fig. 8 (left)) that integrates SAMURAI and SPICE to enable accurate RTN analysis of SRAMs. We have implemented each step of this methodology and obtained results showing that SAMURAI does indeed accurately predict non-stationary RTN effects in SRAMs. We have also validated SAMURAI against analytical results known for stationary RTN.

II. THE RTN GENERATION MECHANISM: FROM TRAPS TO MARKOV CHAINS

A. Origins of RTN: Traps in the dielectric

As mentioned before, RTN is caused by the random capture and release of electrons by traps located in the MOS oxide layer (as shown in Fig. 6). At any given moment, an oxide trap can be in one of two possible *states*, (a) *filled* (*i.e.*, the trap has captured an electron from the inversion layer), or (b) *empty* (*i.e.*, the trap has released any previously captured electron back into the inversion layer). For a given trap, the evolution of its state over time (between *filled* and *empty*) is inherently random, *i.e.*, it is a stochastic process. The parameters of this stochastic process depend on three factors: (a) the vertical distance y_{tr} of the trap from the oxide-semiconductor interface, (b) the trap energy level E_{tr}, and (c) the instantaneous gate bias $V_{gs}(t)$ of the device [6]. It is this dependence on $V_{gs}(t)$ that makes the trap statistics, and hence the induced RTN, non-stationary.

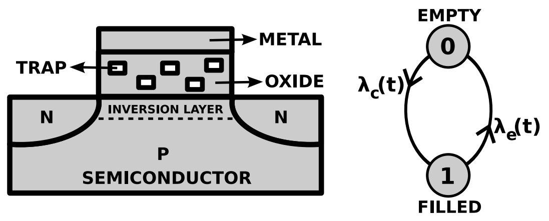

Fig. 6. Left: Traps in a MOS transistor's oxide layer. Right: A time-inhomogeneous two-state Markov chain model for a single trap.

B. Traps as time-inhomogeneous two-state Markov chains

Given that a trap is empty (filled) at time t, the probability that it will become filled (empty) by time $t+dt$ (*i.e.*, it will capture (emit) an electron in the small time interval dt) is given by $\lambda_c(t)dt$ ($\lambda_e(t)dt$), where $\lambda_c(t)$ ($\lambda_e(t)$) is a time-varying function called the *capture (emission) propensity* of the trap.

The above stochastic model governing the activity of a single trap can be described by a two-state time-inhomogeneous Markov chain, as shown in Fig. 6 (right). The two states in this Markov chain are designated 0 (empty) and 1 (filled), while transitions between states are labelled by the corresponding propensity functions.

The functions $\lambda_c(t)$ and $\lambda_e(t)$ depend on the instantaneous bias $V_{gs}(t)$, and also on the trap characteristics y_{tr} and E_{tr}. Detailed equations describing these dependencies can be found in [6], from which we obtain:

$$\lambda_c(t) + \lambda_e(t) = \frac{1}{\tau_0 e^{\gamma y_{tr}}} \tag{1}$$

$$\beta(t) = \lambda_e(t)/\lambda_c(t) = g\ e^{\frac{E_T - E_F}{kT}} \tag{2}$$

$$[\text{where } (E_T - E_F)|_t = \text{function}(E_{tr}, y_{tr}, V_{gs}|_t, \text{device parms})]$$

From the above equations, it is seen that the *sum* $\lambda_c + \lambda_e$ is constant with time, depending only on y_{tr}, the time constant τ_0 for traps at the silicon interface and the tunnelling coefficient γ. However, the *ratio* $\beta = \lambda_e/\lambda_c$ at time t is a complex function (whose exact form is given in [6]) of the instantaneous bias $V_{gs}|_t$. Because the *ratio* β is time-varying, the trap statistics are non-stationary. (Here g is the trap degeneracy factor, k is the Boltzmann constant and T is the temperature.)

C. From trap occupancies to RTN currents

The evolution of the state of *a single trap* is governed by the stochastic model above. In a MOS transistor, there may exist *multiple such traps*. Given a *trap occupancy function* for the device, *i.e.*, a description of how the state of each trap evolves over time, detailed models exist for predicting the noise current $I_{RTN}(t)$ in the device. For example, one model (currently used by SAMURAI) is the following equation [19]:

$$I_{RTN}(t) = \frac{I_d(t)}{WLN(t)}\ N_{filled}(t) \tag{3}$$

where I_d is the nominal drain current (without RTN), W and L are the device dimensions, N is the number density of charge carriers and $N_{filled}(t)$ is the number of device traps filled at time t (which can be calculated from the trap occupancy function). More complex models have also been suggested (*e.g.*, [20]) which, if needed, can be incorporated into SAMURAI just as easily.

III. THE SAMURAI CORE: RTN TRACE GENERATION BY MARKOV UNIFORMISATION

The previous section described how to compute the RTN current *given the trap occupancy function*. This section describes a technique for *computing the trap occupancy function*.

For a device with multiple traps, each trap can be thought of as a separate two-state time-inhomogeneous Markov chain. In order to generate realistic RTN traces under time-varying bias conditions, SAMURAI carries out non-stationary stochastic simulation (Algorithm 1) of each of these Markov chains. The trap occupancy function thus computed is used to generate a realistic RTN trace $I_{RTN}(t)$.

Algorithm 1: Non-stationary RTN generation in SAMURAI

```
Input: Trap profile, Bias $\{V_{gs}(t),\ I_d(t)\ \ldots\}$, $t_0$, $t_f$
Output: Realistic $I_{RTN}(t)$ trace in time interval $[t_0,t_f]$
1  foreach trap tr in the device do
2      compute $\lambda_c(t)$, $\lambda_e(t)$, $t \in [t_0,t_f]$, for tr (use Eq. (1), (2));
3      $\lambda^* = \lambda_c(t_0) + \lambda_e(t_0)$;
4      curr_time = $t_0$; curr_state = tr.init_state;
5      times = [curr_time]; states = [curr_state];
6      while curr_time $< t_f$ do
7          next_cand_time = curr_time + exprand($1/\lambda^*$);
8          curr_time = next_cand_time;
9          if curr_time $> t_f$ then break;
10         if curr_state == 1 then
11             $\lambda_{next} = \lambda_e$(curr_time)
12         else
13             $\lambda_{next} = \lambda_c$(curr_time)
14         end
15         bool change_the_state = rand() $< \lambda_{next}/\lambda^*$;
16         if change_the_state then
17             times.append(curr_time);
18             states.append(curr_state);
19             curr_state = (curr_state == 1) ? 0 : 1;
20             times.append(curr_time);
21             states.append(curr_state);
22         end
23     end
24     trap_occupancy[tr] = [times, states];
25 end
26 compute $I_{RTN}(t)$ from trap_occupancy[tr] (use Eq. (3))
```

Algorithm 1 takes as input, (a) the *trap profile* of the device (*i.e.*, the position y_{tr} and energy E_{tr} of each trap), and (b) the time-varying bias conditions. It produces as output an $I_{RTN}(t)$ trace whose statistics are *exactly identical* to the time-varying RTN statistics under the non-stationary model of the previous section.

Briefly, the algorithm works by generating *more* trap activity than necessary, and then *discarding* some of the generated activity so that the time-varying trap statistics are exactly preserved. Line 3 computes λ^*, which is an upper bound on the functions $\lambda_c(t)$ and $\lambda_e(t)$. In each iteration of the while loop (line 6), a *candidate event* is generated (line 7) corresponding to a *stationary* two-state Markov chain with both propensities set to λ^*. Thus, the original non-stationary Markov chain is first *uniformised* into a stationary (but high rate) Markov chain. In line 15, a probabilistic decision is made to either *keep* or *discard* the generated event, which exactly restores the non-stationarity of the original Markov chain. That this algorithm exactly preserves the original Markov chain's non-stationarity is proved in [11]–[13].

Although Algorithm 1 generates $I_{RTN}(t)$ traces only for a *single device*, it can be straightforwardly extended to investigate the effect of RTN on *entire SRAM cells* (as we do in the next section).

IV. RESULTS

In this section we present two kinds of results, (a) *validation results* that demonstrate excellent agreement between SAMURAI's predictions and analytical expressions known for stationary RTN, and (b) *simulation results* showing that SAMURAI can accurately predict the effects of non-stationary RTN in SRAM cells.

A. Validation results

SAMURAI is primarily intended for non-stationary RTN analysis under arbitrarily time-varying bias conditions. Although analytical expressions are not available for such a general case, they are known for the restricted constant bias case [3], [5]. Here we validate SAMURAI against these expressions for a wide range of trap configurations.

- We run three validation experiments, using typical values for the 3 parameters that affect the trap capture/release statistics, namely, V_{gs}, E_{tr}, and y_{tr}. In each experiment, we fix two of these parameters and sweep the third over an appropriate range. Hence we generate a variety of trap configurations, which are then simulated under constant gate bias using Algorithm 1.
- Algorithm 1 returns a trace $I_{RTN}(t)$. From this trace, we numerically estimate the autocorrelation function $R(\tau) = E[I_{RTN}(t)I_{RTN}(t+\tau)]$.
- We also translate the above time-domain results into frequency domain by computing the stationary power spectral density $S(f)$ numerically from $R(\tau)$.
- We plot the above waveforms $R(\tau)$ and $S(f)$ alongside analytical expressions obtained from [3], [5]. To get an idea of the relative importance of RTN, we also plot the stationary power spectral density of thermal noise in the device using the model $S_{thermal}(f) = \frac{8}{3}kTg_m$ (where k is the Boltzmann constant, T is the temperature and g_m is the device transconductance at the applied bias).

The results are presented in Fig. 7 (a to f). In all these plots, τ is measured in seconds, $R(\tau)$ in A^2, all frequencies are in Hz and all spectral densities are in A^2/Hz. From Fig. 7, it is seen that the RTN traces predicted by SAMURAI closely match analytical expressions in both the time domain (autocorrelation plots (a)-(c)) and the frequency domain (spectral density plots (d)-(f)).

B. SRAM simulation results

Fig. 8 (left) shows a flowchart illustrating our methodology for analysing non-stationary RTN in SRAMs. This methodology combines SAMURAI with SPICE, resulting in an accurate, trap-level, simulation-driven strategy for SRAM design in the presence of RTN.

- First we simulate the SRAM cell (on a test pattern of reads and writes) without RTN in SPICE. This enables the generation of time-varying biasses (to be used as input to SAMURAI).
- Next, we use SAMURAI to generate RTN traces for each transistor in the SRAM cell, under the biasses obtained from SPICE. In addition to the biasses, this step requires a trap profile for each device, which is either obtained from measurement data [7] or generated using statistical trap profiling models proposed in the literature [6].
- We model the RTN traces generated above as current sources between the drain and source of each device (similar to Fig. 4 (right)). We again carry out SPICE simulation of the SRAM cell (on the same test pattern), this time including the I_{RTN} of each transistor.
- If the second SPICE simulation predicts write errors or unacceptable slowdown in SRAM operation, it is immediately clear that the SRAM cell is compromised due to RTN (either V_{dd} must be increased or the SRAM cell must be re-designed). Otherwise, the analysis is repeated with a new test pattern (or a conclusion is reached that the SRAM cell is indeed robust to RTN).

In Fig. 8 (right), we have implemented each step of the above methodology (in 90nm technology using BSIM-4 device models and SpiceOPUS [21] for circuit simulation) and presented the results in 5 plots (labelled (a)-(e)) alongside the flowchart, with arrows pointing from the flowchart stages to the relevant plots. We have illustrated the entire methodology using the bit pattern [1,1,0,1,0,1,0,0,1] written to the SRAM cell. As seen from plot (a), this bit pattern is successfully written to the SRAM cell if

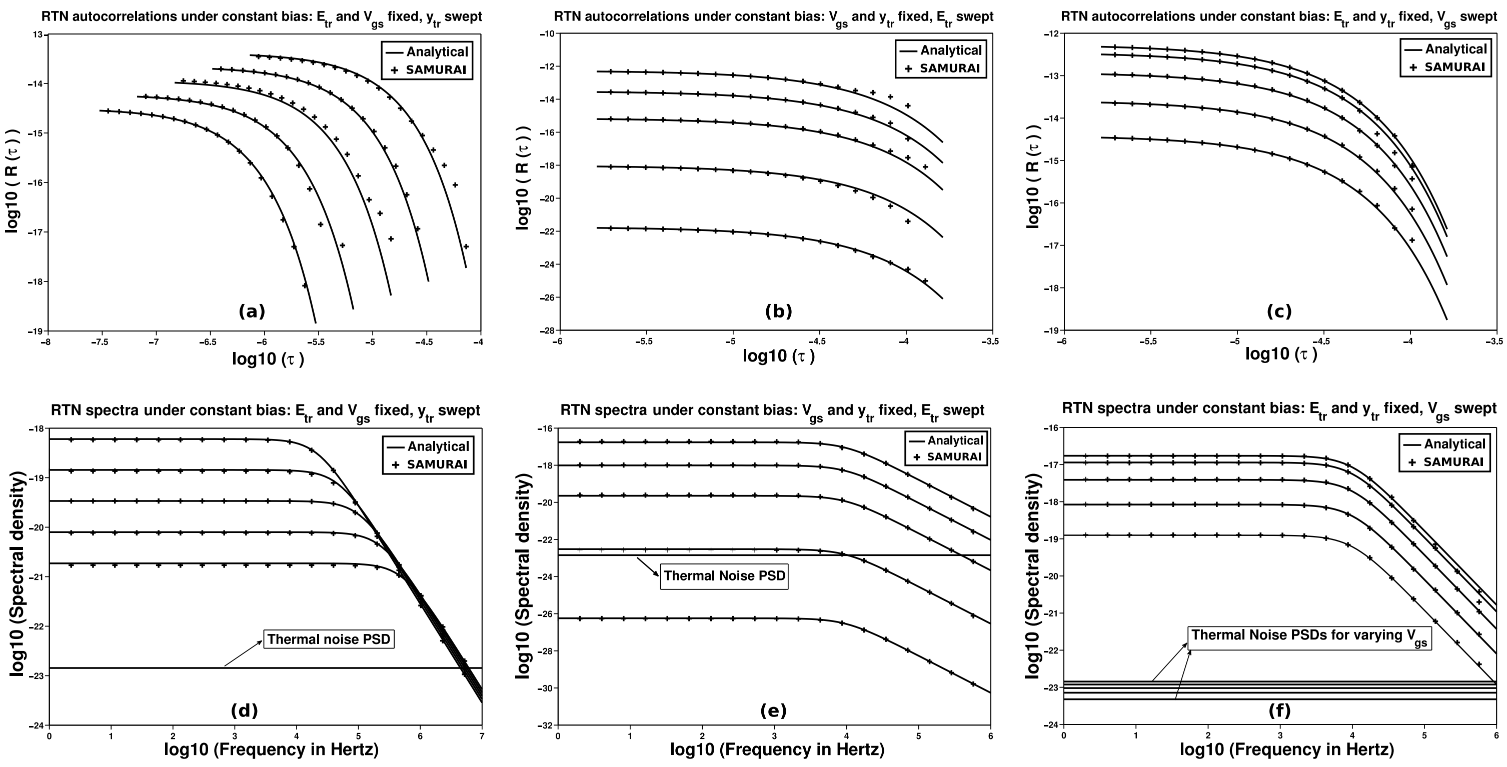

Fig. 7. Plots showing that the RTN traces generated by SAMURAI closely match analytical predictions in both the time domain (autocorrelation plots (a)-(c)) and the frequency domain (spectral density plots (d)-(f)).

there is no RTN.

Plots (b) and (c) show the trap occupancy functions returned by SAMURAI for transistors M5 and M6 respectively (here we have used the statistical trap profiling model proposed in [6]). As is to be expected [3], the transistor M5 (whose gate voltage is Q) exhibits a high degree of trap activity when Q is high, but very little trap activity when Q is low. The opposite is true for transistor M6, whose gate voltage is $\overline{Q}$. Thus, SAMURAI is able to accurately predict the time-varying (non-stationary) statistics of trap activity under rapidly switching bias conditions. (For illustrative purposes, an exploded view of the trap activity during a small time interval from each of these plots is also shown.)

Plot (d) shows the RTN trace generated by SAMURAI for the transistor M2. Plot (e) shows a SPICE simulation of the SRAM cell (on the same bit pattern) after including the I_{RTN} current sources for each transistor. Here we wanted to demonstrate that SAMURAI is indeed capable of predicting non-stationary effects such as write errors in SRAM cells. However, such failures are extremely rare events. Therefore, for illustrative purposes, we have scaled the I_{RTN} trace of each transistor by a factor of 30, which immediately produced a write error (as seen from plot (e)). In more deeply scaled CMOS technologies (*e.g.*, 22nm), such artificial RTN scaling would not be necessary. Moreover, if other sources of variability (*e.g.*, local/global parameter variations, NBTI *etc.*) are taken into account, the incremental effect of RTN would be sufficient to produce a bit error (without artificial scaling). Also, instead of scaling I_{RTN}, we note that a similar effect can be achieved by adopting accelerated RTN testing techniques for SRAMs (such as the one outlined in [14]). That is, SAMURAI should be run on the SPICE response predicted for the SRAM cell under the biasses suggested by accelerated testing techniques.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented SAMURAI, a computational technique that enables accurate, trap-level, non-stationary analysis of RTN in SRAMs. The core of SAMURAI is a procedure called Markov Uniformisation, which extends stochastic simulation ideas from the biological community to the RTN characterisation problem in SRAMs. Starting from a two-state time-inhomogeneous Markov chain model for each device trap, we have demonstrated that SAMURAI is capable of generating non-stationary RTN traces for entire circuits under (a) arbitrary trap populations, and (b) arbitrarily time-varying bias conditions. To the best of our knowledge, SAMURAI is the first computational tool that incorporates sophisticated, trap-level, stochastic RTN models into a methodology for circuit level RTN characterisation. Where analytical expressions are available, SAMURAI closely matches their predictions. And where analytical expressions are not available, SAMURAI still enables accurate RTN characterisation by generating realistic RTN traces. We have also evolved a methodology that integrates SAMURAI and SPICE to conduct RTN analysis of SRAMs. Our implementation of this methodology demonstrates that SAMURAI can indeed predict non-stationary effects such as RTN-induced SRAM write errors.

Against the above accomplishments, we now identify four directions for future research:

1. *Bi-directionally coupled RTN simulation.* Throughout this paper, we have assumed that the biasses for RTN trace generation can be pre-computed by a SPICE simulation. However, in reality, RTN cannot be isolated from the rest of the circuit, *i.e.*, both RTN and the circuit states evolve together, with RTN modulating the circuit voltages/currents and the circuit simultaneously modulating the stochastic processes governing RTN generation, thereby forming a bi-directionally coupled system. For the future, our aim is to accurately simulate such "higher order" effects associated with RTN.
2. *Accounting for other sources of variability.* In this paper, we have discussed the stand-alone impact of RTN on SRAMs. In reality, RTN occurs on top of other non-idealities such as static noise, local/global parameter variations, NBTI *etc.* In future, we would like to properly account for all these variabilities within our simulation tool.
3. *Statistical analysis of RTN for entire SRAM arrays.* In this paper, we

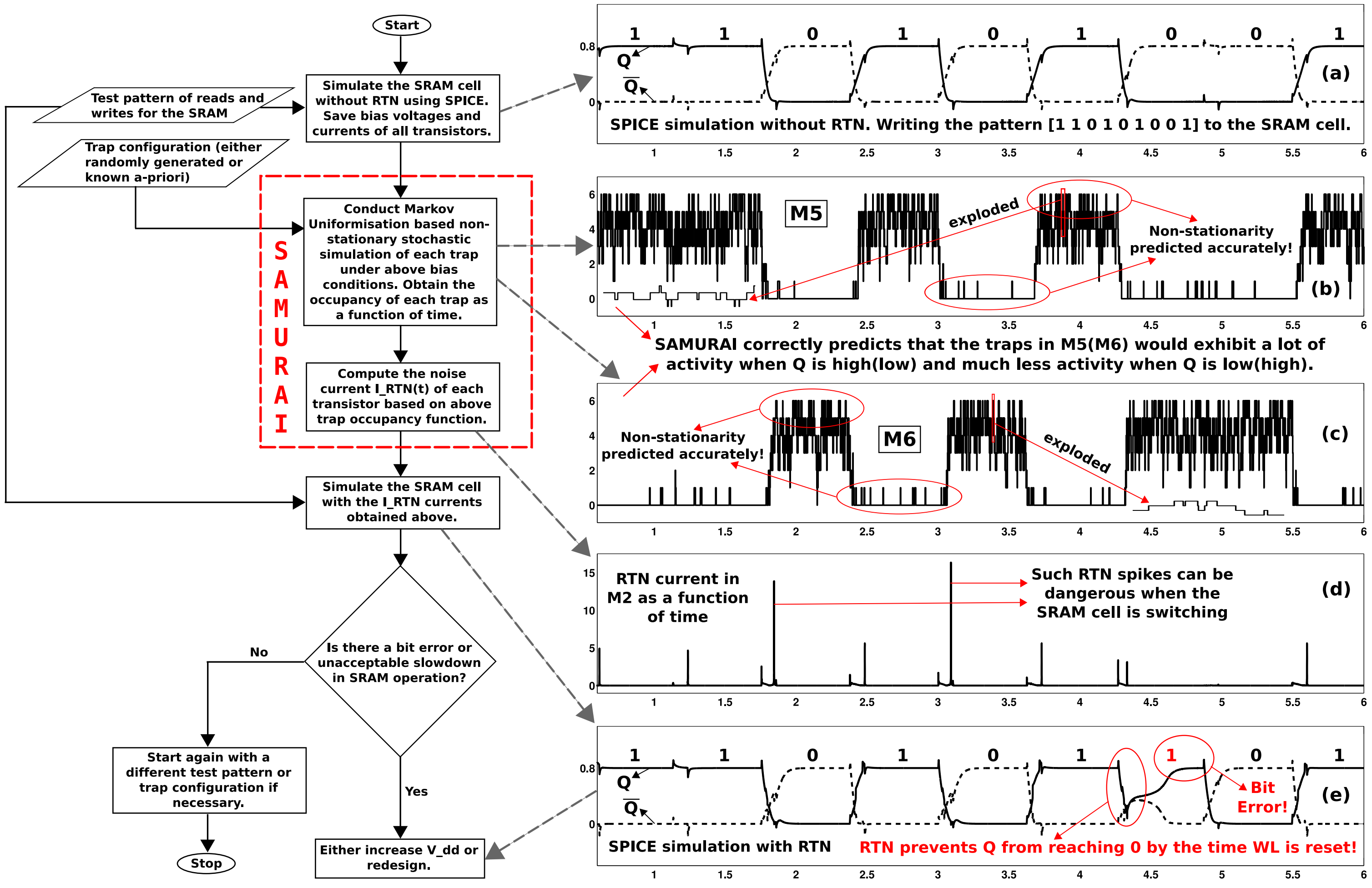
Start
Test pattern of reads and writes for the SRAM
Simulate the SRAM cell without RTN using SPICE. Save bias voltages and currents of all transistors.
Trap configuration (either randomly generated or known a-priori)
SAMURAI
Conduct Markov Uniformisation based non-stationary stochastic simulation of each trap under above bias conditions. Obtain the occupancy of each trap as a function of time.
Compute the noise current I_RTN(t) of each transistor based on above trap occupancy function.
Simulate the SRAM cell with the I_RTN currents obtained above.
Is there a bit error or unacceptable slowdown in SRAM operation?
No
Yes
Start again with a different test pattern or trap configuration if necessary.
Stop
Either increase V_dd or redesign.
1 1 0 1 0 1 0 0 1
Q
Q̄
(a)
SPICE simulation without RTN. Writing the pattern [1 1 0 1 0 1 0 0 1] to the SRAM cell.
M5
exploded
Non-stationarity predicted accurately!
(b)
SAMURAI correctly predicts that the traps in M5(M6) would exhibit a lot of activity when Q is high(low) and much less activity when Q is low(high).
Non-stationarity predicted accurately!
M6
exploded
(c)
RTN current in M2 as a function of time
Such RTN spikes can be dangerous when the SRAM cell is switching
(d)
1 1 0 1 0 1 1 0 1
Bit Error!
(e)
SPICE simulation with RTN
RTN prevents Q from reaching 0 by the time WL is reset!

Fig. 8. Left: Flowchart illustrating our methodology for non-stationary RTN analysis of SRAMs. Right: Plots showing our implementation of this methodology. All plots have the same X-axis (time in ns). On the Y-axis, plots (a,e) show voltages (in V), plots (b,c) show the number of filled traps and plot (d) shows current in μA.

have considered the effect of RTN on a single SRAM cell. We view this as the first step towards predicting the bit-error impact of RTN on entire SRAM arrays, which are made up of thousands of SRAM cells that are subject to local and global parameter variations.

4. *Applications beyond SRAMs.* RTN has been shown to adversely affect many circuits other than SRAMs. For instance, RTN is thought to be responsible for Variable Retention Time (VRT) in DRAMs [22], [23]. RTN is also known to impact ring oscillators [3]. We also conjecture that RTN causes cycle slipping in Phase Locked Loops (PLLs). In future, we would like to extend SAMURAI to conduct RTN analysis for all these different circuits.

Acknowledgements: We would like to thank Yasumasa Tsukamoto of Renesas Electronics Corp. for Fig. 2 (and many useful discussions, besides). Support from the Semiconductor Research Corporation (SRC, task 1836.021) is also gratefully acknowledged. We would like to thank David Yeh of SRC, in particular, for his encouragement of this work.

References

[1] Y. Tsukamoto, S.O. Toh, C. Shin, A. Mairena, T.J.K. Liu, and B. Nikolic. Analysis of the relationship between random telegraph signal and negative bias temperature instability. In *Proceedings of the IEEE International Reliability Physics Symposium*, pages 1117–1121, 2010.

[2] J.S. Kolhatkar. *Steady-state and cyclo-stationary RTS noise in MOSFETS*. PhD thesis, The University of Twente, The Netherlands, 2005. http://doc.utwente.nl/48261/1/thesis-Kolhatkar.pdf.

[3] H. Tian and A. El Gamal. Analysis of 1/f noise in switched MOSFET circuits. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 48(2):151–157, 2001.

[4] A. van der Ziel. *Noise in solid state devices and circuits*. Wiley Interscience, 1976.

[5] M.J. Kirton and M.J. Uren. Noise in solid-state microstructures: A new perspective on individual defects, interface states and low-frequency 1/f noise. *Advances in Physics*, 38(4):367–468, 1989.

[6] M.V. Dunga. *Nanoscale CMOS modeling*. PhD thesis, The University of California, Berkeley, 2008. http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-20.html.

[7] S. Lee, H.J. Cho, Y. Son, D.S. Lee, and H. Shin. Characterisation of oxide traps leading to RTN in high-K and metal gate MOSFETS. In *Proceedings of the IEEE International Electron Devices Meeting*, pages 763–766, 2009.

[8] G.I. Wirth, J. Koh, R. da Silva, R. Thewes, and R. Brederlow. Modelling of statistical low frequency noise of deep sub-micron MOSFETS. *IEEE Transactions on Electron Devices*, 52(7):1576–1588, 2005.

[9] D.T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[10] Y. Ye, C.C. Wang, and Y. Cao. Simulation of Random Telegraph Noise with 2-stage equivalent circuit. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, 2010.

[11] P. Heidelberger and D.M. Nicol. Conservative parallel simulation of continuous time Markov chains using uniformisation. *IEEE Transactions on Parallel and Distributed Systems*, 4(8):906–921, 1993.

[12] A.P.A. van Moorsel and K. Wolter. Numerical solution of non-homogeneous Markov processes through uniformisation. In *Proceedings of the Twelfth European Multiconference on Simulation*, pages 710–717, 1998.

[13] J.G. Shanthikumar. Uniformisation and hybrid simulation/analytic models of renewal processes. *Operations Research*, 34(4):573–580, 1986.

[14] S.O. Toh, Y. Tsukamoto, Z. Guo, L. Jones, T.J.K. Liu, and B. Nikolic. Impact of random telegraph signals on Vmin in 45nm SRAM. In *Proceedings of the IEEE International Electron Devices Meeting*, pages 767–770, 2009.

[15] L. Brusamarello, G.I. Wirth, and R. da Silva. Statistical RTS model for digital circuits. *Microelectronics Reliability*, 49(9-11):1064–1069, 2009.

[16] K. Takeuchi, T. Nagumo, K. Takeda, S. Asayama, S. Yokogawa, K. Imai, and Y. Hayashi. Direct observation of RTN-induced SRAM failure by accelerated testing and its application to product reliability assessment. In *Proceedings of the IEEE International Symposium on VLSI Technology*, pages 189–190, 2010.

[17] J. Pahle. Biochemical simulations: stochastic, approximate stochastic and hybrid approaches. *Briefings in Bioinformatics*, 10(1):53–64, 2009.

[18] A.M. Kierzek. STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm. *Bioinformatics*, 18(3):470–481, 2002.

[19] A. van der Ziel. Unified presentation of 1/f noise in electron devices: fundamental 1/f noise sources. *Proceedings of the IEEE*, 76(3):233–258, 1988.

[20] K.K. Hung, P.K. Ko, C. Hu, and Y.C. Cheng. A physics-based MOSFET noise model for circuit simulators. *IEEE Transactions on Electron Devices*, 37(5):1323–1333, 1990.

[21] http://www.spiceopus.si/.

[22] P.J. Restle, J.W. Park, and B.F. Lloyd. DRAM variable retention time. In *Proceedings of the IEEE International Electron Devices Meeting*, pages 807–810, 1992.

[23] T. Umeda, K. Okonogi, K. Ohyu, S. Tsukada, K. Hamada, S. Fujieda, and Y. Mochizuki. Single silicon vacancy-oxygen complex defect and variable retention time phenomenon in DRAMs. *Applied Physics Letters*, 88(25):253504(1–3), 2006.