Impact of NBTI on SRAM Read Stability and Design for Reliability

Sanjay V. Kumar, Chris H. Kim, and Sachin S. Sapatnekar
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455

***Abstract*— Negative Bias Temperature Instability (NBTI) has the potential to become one of the main show-stoppers of circuit reliability in nanometer scale devices due to its deleterious effects on transistor threshold voltage. The degradation of PMOS devices due to NBTI leads to reduced temporal performance in digital circuits. We have analyzed the impact of NBTI on the read stability of SRAM cells. The amount of degradation in Static Noise Margin (SNM), which is a measure of the read stability of the 6-T SRAM cell has been estimated using Reaction-Diffusion (R-D) model. We propose a simple solution to recover the SNM of the SRAM cell using a data flipping technique and present the results simulated on BPTM 70nm and 100nm technology. We also compare and evaluate different implementation methodologies for the proposed technique.**

***Index Terms* : Negative Bias Temperature Instability (NBTI), SRAM, Cache, Static Noise Margin (SNM), Reaction-Diffusion (R-D) Model.**

I. INTRODUCTION

The generation of interface traps under negative bias conditions (V_{gs}=–V_{dd}) at elevated temperatures in PMOS transistors is called *Negative Bias Temperature Instability* (NBTI). NBTI has proven to be a growing threat to circuit reliability in nanometer scale technology [1]–[8].

Interface traps are formed due to crystal mismatches at the Si-SiO_2 interface. During oxidation of Si, most of the tetrahedral Si atoms bond to oxygen. However, some of the atoms bond with hydrogen, leading to the formation of weak Si-H bonds (Fig. 1(i)-(ii)). When a PMOS transistor is biased in inversion, the holes in the channel dissociate these Si-H bonds, thereby generating interface traps (Fig. 1(iii)). Interface traps (interface states) are electrically active physical defects with their energy distributed between the valence and the conduction band in the Si band diagram [1]. They are manifested as an increase in absolute PMOS transistor threshold voltage ($|V_{tp}|$) and a reduction in absolute I_{on} current of PMOS devices, thereby making them slower. An increase in $|V_{tp}|$ not only leads to reduced temporal performance but may also cause reliability issues and potential device failure.

The effects of NBTI on digital CMOS circuits have been analyzed by authors in [2], [4], [5], [7]–[9]. While the temporal degradation of static CMOS circuits can be offset by transistor up-sizing during design [8] (to account for the decrease in speed of the PMOS devices due to NBTI), memory circuits pose a much greater challenge. Area-speed trade-off solutions do not work efficiently for SRAM arrays since area is a much greater concern in memory design as compared to digital CMOS or analog circuit design. Previous literature that dealt with the effect of NBTI on SRAM cells, such as [2] and [5], measured the extent of degradation of SNM due to NBTI with respect to a reduction in V_{dd}. However, our work, the first of its kind, focuses on the temporal SNM degradation of SRAM cells due to NBTI. We perform transistor level simulations using BPTM 70nm and 100nm models [10], [11], with V_{tp} shifted model files to simulate the effect of NBTI on PMOS devices. Simulations performed using this methodology of absolute shift in V_{tp} show that the SNM worsens by about 8-9% after 10^8 seconds ($\approx$ 3 years) for 100nm and 70nm devices.

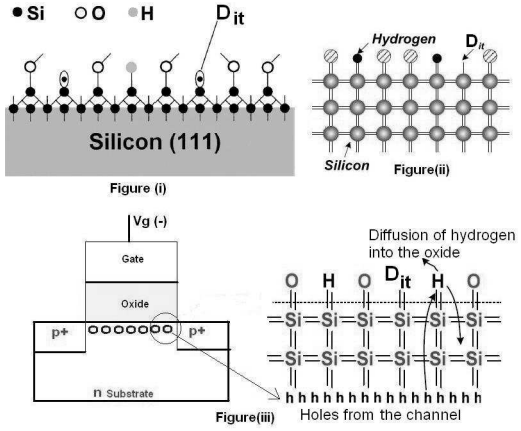

Fig. 1. Schematic description showing the generation of interface traps when a PMOS transistor is biased in inversion. Figure (i) shows the 3-D structure of Si at the $Si-SiO_2$ interface in the 111 crystal orientation. N_{it} is the site containing an unsaturated electron (crystal mismatch) leading to the formation of an interface trap. Figure (ii) shows the $Si-SiO_2$ interface in 2-D along with the $Si-H$ bonds and the interface traps. Figure (iii) shows the dissociation of Si-H bonds by the holes when the PMOS device is biased in inversion and the diffusion of hydrogen into the oxide, thereby generating an interface trap [1].

While the application of a continuous negative bias to the gate of the PMOS transistor degrades its temporal performance, removal of the bias helps anneal some of the interface traps generated, leading to a partial recovery of the threshold voltage. The process of degradation and recovery is successfully analyzed using the Reaction-Diffusion (R-D) model. We utilize this physical phenomenon to develop a novel solution to overcome the effect of NBTI by flipping the contents of the SRAM cell periodically. This ensures that the PMOS devices are subjected to periods of alternate stress and relaxation (as opposed to continuous stress) allowing dynamic recovery of threshold voltage. We present results obtained through simulations based on the R-D model, which indicate that about **30% of read stability** (measured in terms of SNM)

can be restored through cell flipping. Hardware and software implementations for this novel methodology have also been discussed.

The rest of the paper is organized as follows. In Section II, we describe the manifestation of NBTI on PMOS threshold voltage. The simulation methodology and the results obtained are explained in Section III. Section IV focuses on the novel solution to recover NBTI affected SRAM cells while Section V provides a qualitative overview of the implementation followed by conclusions in Section VI.

II. Effect of NBTI on PMOS Transistor Threshold Voltage

Several models have been proposed to physically explain the mechanism of NBTI based on the activation energy of electro-chemical reactions [1], [3], [4], [6], [12]–[20]. We have used the Reaction Diffusion (R-D) Model, which has been widely used by authors in [3], [6], [17], [18], [21] on account of its ability to successfully explain the physics of interface trap generation mechanism. According to the R-D model, the rate of generation of interface traps (N_{IT}) initially depends on the rate of dissociation of the Si-H bonds, which is controlled by the forward rate constant (k_f) and the local self-annealing process which is governed by the rate constant (k_r) [3], [17], [18]. This constitutes the reaction phase in the R-D model. Thus,

$$\frac{dN_{IT}}{dt} = k_f[N_0 - N_{IT}] - k_r N_{IT} N_{H(0)} \tag{1}$$

where N_0 is the maximum density of $Si-H$ bonds and $N_{H(0)}$ is the hydrogen density at the interface.

After sufficient trap generation, the rate of generation of traps is limited by the diffusion of hydrogen and follows the equations,

$$\begin{aligned} \frac{dN_{IT}}{dt} &= D\frac{dN_H}{dx} \\ \frac{dN_H}{dt} &= D\frac{d^2N_H}{dx^2} \end{aligned} \tag{2}$$

where D is the diffusion of hydrogen species. This constitutes the diffusion phase in the R-D model. The above two equations can be solved for the steady state condition $\frac{dN_{IT}}{dt} = 0$ to obtain an expression for N_{IT} as derived in [3] and [17]:

$$N_{IT} = \sqrt{\frac{k_f N_0}{k_r}}(Dt)^{0.25}. \tag{3}$$

This is in accordance with the power law model which states that the generation of interface traps follows a t^α relationship where α is between 0.17 and 0.3 [3], [5], [17], [22], [23]. The validity of the R-D model has been confirmed in [17] where experimental results conform to values obtained by simulation.

The threshold voltage for a PMOS transistor from [1] and [24] is given by

$$V_T = V_{FB} - 2\phi_b - \frac{|Q_B|}{C_{ox}} \tag{4}$$

where $\phi_b = (kT/q)ln(N_D/n_i)$ and Q_B is the depletion region charge density. V_{FB} is the flat band voltage given by

$$V_{FB} = \phi_{ms} - \frac{Q_f}{C_{ox}} - \frac{Q_{it}(\phi_s)}{C_{ox}} \tag{5}$$

and

$$Q_{it}(\phi_s) = qN_{IT} \tag{6}$$

where Q_f denotes the fixed charge density, Q_{it} denotes the interface trapped charge density, C_{ox} is the oxide capacitance per unit area, ϕ_{ms} is the work-function difference, and ϕ_s is the surface potential.

The generation of interface traps due to NBTI causes a shift in the transistor threshold voltage given by

$$\Delta V_{tp} = -\frac{qN_{IT}}{C_{ox}}. \tag{7}$$

However, in order to account for the fact that trap generation causes mobility degradation which further leads to threshold voltage degradation [4], the above equation is modified as

$$\Delta V_{tp} = -\frac{(m+1)qN_{IT}}{C_{ox}} \tag{8}$$

where m is a measure of the additional V_{tp} degradation caused due to mobility degradation [8].

Equations (3) and (8) can be used to determine the threshold voltage at any given point of time. These values can be used to perform transistor level simulations for a 6-T SRAM cell using a V_{tp} shifted model file, as described in the next section.

III. Impact of NBTI on 6-T SRAM Cell

In this section we present the simulation results on the SRAM cell shown in Fig 2. We assume that NBTI causes the threshold voltage of the PMOS transistor modeled using BPTM 70nm and 100nm technology files [10], [11] to change from its nominal value by 10% after 10^6 seconds (≈ 11.6 days). This assumption is based on the simulation results obtained by the authors in [8] using (8). Accordingly, the value of V_{tp} is changed from its nominal value of -0.22V to -0.242V for 70nm in the model file (from -0.303 to -0.3333V for 100nm). We have performed simulations using HSPICE on the 6-transistor SRAM cell. The three main parameters analyzed are read-delay, write-delay and Static Noise Margin (SNM) which is a measure of read stability. The results are tabulated in Tables I and II for both 100nm and 70nm devices.

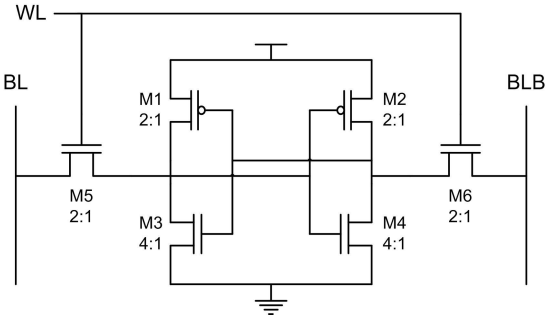

Fig. 2. Six transistor SRAM cell

It can be seen from the tables that the read delay is virtually unaffected, the write delay improves marginally, while the SNM of the SRAM cell decreases due to NBTI. The SNM degradation as a function of $|V_{tp}|$ is plotted in Fig. 3 (i) and

TABLE I

PERFORMANCE DEGRADATION (AFTER $10^6 s$) FOR A 100NM SRAM CELL ($V_{tp} = -0.303V$) AND 70NM SRAM CELL ($V_{tp} = -0.22V$) WITH $V_{dd} = 1V$ AT $T = 110°C$

	100nm cell		70nm cell	
parameter	nominal	NBTI affected	nominal	NBTI affected
Read delay	187.6ps	187.6ps	186.6ps	186.1ps
Write delay	45.69ps	45.31ps	40.95ps	40.68ps
SNM	0.1278V	0.1254V	0.1007V	0.0987V

TABLE II

SNM DEGRADATION FOR 100NM AND 70NM SRAM CELLS WITH $V_{dd} = 1V$ AT $T = 110°C$

	100nm cell		70nm cell	
time (s)	V_{tp} (V)	SNM (V)	V_{tp} (V)	SNM (V)
0	-0.303	0.1278	-0.22	0.1007
1	-0.3039	0.1277	-0.2207	0.1007
10^1	-0.3047	0.1277	-0.2212	0.1007
10^2	-0.3060	0.1277	-0.2222	0.1007
10^3	-0.3083	0.1277	-0.2239	0.1007
10^4	-0.3125	0.1273	-0.2270	0.1005
10^5	-0.3200	0.1267	-0.2324	0.0999
10^6	-0.3333	0.1254	-0.2420	0.0987
10^7	-0.3569	0.1227	-0.2591	0.0964
10^8	-0.3988	0.1174	-0.2896	0.0915

(ii) for the 100nm and 70nm devices respectively. It can be seen from Table II that NBTI increases the absolute threshold voltage which leads to gradual reduction in the SNM. This can lead to read stability issues and can potentially cause failures.

IV. RECOVERING STATIC NOISE MARGIN IN SRAM CELLS

The generation of interface traps due to negative bias is also accompanied by a process of annealing of these traps when the negative bias applied at the gate is removed [4], [6], [9], [14], [16]–[19], [22], [23]. Thus, if the voltage applied at the gate of the PMOS device is regularly switched, dynamic recovery of threshold voltage occurs and thereby significant amount of performance can be recovered. This concept of performance recovery due to the application of periodic stress and relaxation on the gate of the PMOS device can be used to improve the SNM of the SRAM cell. Due to the topology of SRAM cells, one of the PMOS transistors is always turned on while the other one is turned off. Only one of the PMOS transistors is affected by NBTI if the cell contents are not modified. Hence, if we periodically flip the contents of the cell, the effect can be balanced out. This idea is similar in theory to applying AC stress on the PMOS transistors as opposed to DC stress. It has been experimentally shown in [13], [16]–[18] that the amount of threshold voltage shift in AC conditions is much lower than that for the DC case.

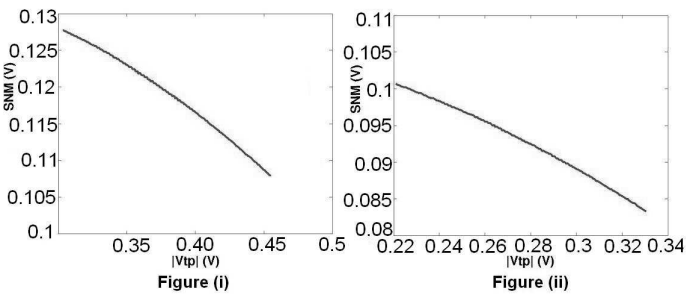

Fig. 3. SNM versus $|V_{tp}|$ for a SRAM cell simulated at V_{dd}=1.0V and $T = 110°C$ using (i)100nm and (ii)70nm BPTM technology.

This reversible phenomenon can be successfully analyzed using the R-D model. If we assume that the PMOS device is under NBTI stress from $t = 0$ to t_0 and goes through a relaxation phase for $t > t_0$, some of the generated traps are annealed. Assuming double sided diffusion, the number of interface traps ($N_{IT}(t)$), at any time $t > t_0$, is given by

$$N_{IT}(t > t_0) = \left(\frac{N_{IT}(t_0)}{1 + \sqrt{\frac{0.5(t-t_0)}{t}}} \right) \tag{9}$$

where $N_{IT}(t_0)$ is the total number of interface traps when stress is removed and relaxation begins. At $t = 2t_0$, $1/3^{rd}$ of the traps are annealed. The R-D model has been used to simulate the stress and relaxation phases in [4], [9], [17], [18], and simulation results concur with experimental results. Fig. 4 plots the interface trap generation versus time for the degradation-recovery case against the case where the PMOS device is subjected to continuous stress assuming that the stress and relaxation periods are both equal to t_0. We specifically choose the stress and relaxation periods to be equal to t_0 since only one of the two PMOS transistors (M1 and M2) in the SRAM cell (Fig. 2) is under NBTI stress at any given point of time (and the other is in relaxation phase), and we would like to balance the effect of NBTI on both M1 and M2.

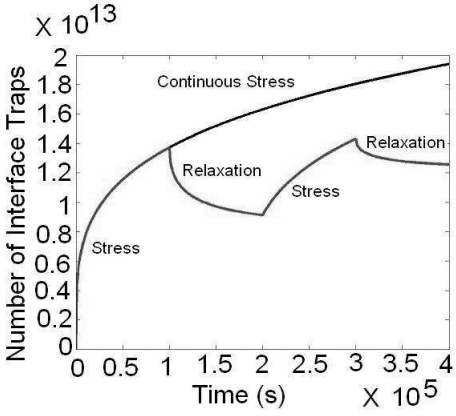

Fig. 4. N_{IT} versus t for two cycles of periodic stress and relaxation with $t_0 = 10^5$ seconds against the case where there is continuous stress.

As can be seen from Table II, NBTI is a fairly slow mechanism and the amount of degradation in SNM is noticeable only after 10^5 seconds (≈ 1.16 days). Hence, it is adequate to flip the contents of the cell at a frequency of once a day. We hereby present simulation results for our SRAM cell, assuming a flipping rate of 10^5 seconds. Using (3), (8) and (9) and the fact that V_{tp} changes by 10% after 10^6 seconds [8], the N_{IT} values, and thereby the V_{tp} values, can be calculated as

TABLE III
STATIC NOISE MARGIN (VOLTS) FOR THE SRAM CELL

	$t = 0$	$t = 10^8$ seconds				
		Non-cell flipping		Cell flipping		
device	SNM	SNM	Δ SNM	SNM	Δ SNM	Recovery
100nm	0.1278	0.1174	-0.0104	0.1205	-0.0703	30%
70nm	0.1007	0.0915	-0.0092	0.0944	-0.0063	29%

a function of time. A plot of $|V_{tp}|$ versus time is shown in Fig. 5 (i) and (ii) for the 100nm and 70nm cells respectively. The V_{tp} values are calculated at different time intervals and a look up table of SNM versus V_{tp} is built by simulating the SRAM at each value of V_{tp}. The SNM can also be plotted as a function of time and the plot for both the cell flipping and non-cell flipping case is shown in Fig. 6 (i) and (ii) for 100nm and 70nm devices respectively. It can be seen from Table III that cell flipping (at an interval of 10^5 seconds) reduces the amount of SNM degradation by 30% for both 100nm and 70nm devices after 10^8 seconds (≈ 3 years).

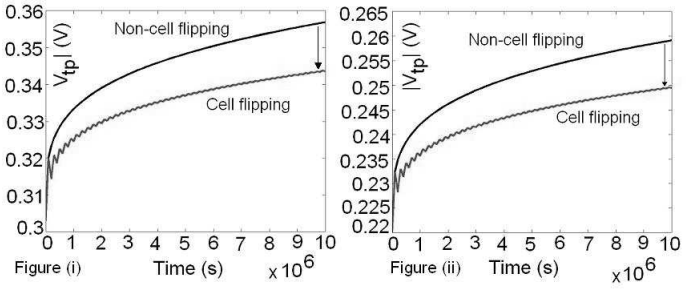

Fig. 5. $|V_{tp}|$ versus t for periodic stress and relaxation with $t_0 = 10^5$ seconds for (i) 100nm and (ii) 70nm SRAM cell.

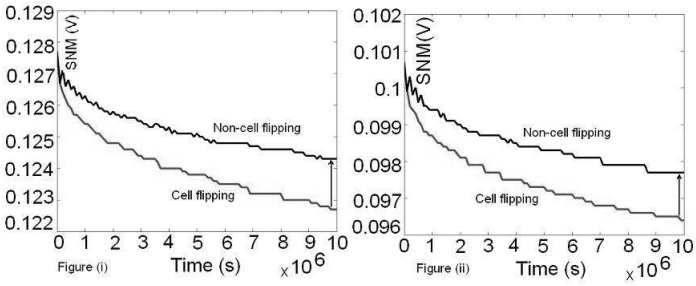

Fig. 6. SNM versus time for cell flipping and non-cell flipping case for (i) 100nm and (ii) 70nm cell.

V. IMPLEMENTATION OF CELL FLIPPING IN SRAM ARRAYS

In this section, we provide an overview of the implementation of cell flipping in SRAM arrays, which are typically present in cache blocks in a processor. The two main aspects of SRAM cell flipping are the ability to flip the contents of all cells periodically, and the ability to read and write data correctly during normal course of operation, and these are explained in the sub-sections below.

A. Periodic flipping of SRAM cells

Flipping the contents of all cells every 10^5 seconds can be performed either through software or hardware.

Software approach: In the software approach, a subroutine is written in the system to interrupt the normal operation of the processor every 10^5 seconds. The subroutine runs from the first addressable location of the SRAM array till the last location and generates an incremental address. The data from the array is first read into the processor registers. This data is inverted and written back to the same address. The address is then incremented and the loop runs till all the contents are flipped. The advantage of this approach is that it has zero hardware overhead and can be programmed into existing processors by writing a subroutine that runs every 10^5 seconds. However, for large caches (say L3 caches) which are far away from the processor, access to a single location may take about 100 clock cycles. To read the data from every address, invert it and write back takes more than 200 clock cycles per location. It is unrealistic to run this subroutine over the entire cache since typically L3 caches are a few megabytes in size.

Hardware approach: In the hardware approach, the SRAM array is embedded with additional hardware and control signals as shown in Fig. 7. A 1 bit control signal (Flip) with two mutually exclusive states to indicate cache data access available to the processor and cache data being flipped is used. During a normal data access in the cache, the address is placed on the address bus and is sent to the address decoder through S2 and the corresponding word lines on S3 are activated. In case of a read operation, the data is read from the SRAM array (S4), and the output of the sense amplifiers is the final read-data sent to the processor. Writes proceed in a similar manner except that the write-data is placed on the write-data bus (S7) and is written into the cache. For cell-flipping, the read-data after the sense amplifiers (S6) is inverted and the negated data (S5) is multiplexed with the actual processor write-data. A counter capable of generating consecutive addresses (S1) is designed such that the successive word lines are activated. The data from the bit lines are read, inverted, placed on the write-data bus and are written back to the same location through S7. The counter clock cycle is equal to the read access time, plus the write access time, plus the overhead in inverting the data and placing it on the write-data bus. This process is continued until all addresses are accessed and the entire data inside the cache is flipped.

The hardware approach is much faster compared to the software approach since the data is flipped locally using inverters and written back through the multiplexers. The total time of flipping depends on the size of the cache only and not its relative distance from the processor, thereby providing maximum benefit for L3 caches. Typically, in a processor system, this flipping operation can be performed when the processor enters *stand-by* mode. This ensures that the normal processor operation is not affected and also helps better maintain cache coherence since the data is not being modified by the processor when it is being flipped internally. Further, since the cache is not accessed by the processor during stand-by mode, the impact of NBTI is most significant since the cache data remains unaltered for long periods of time (due to continuous stress

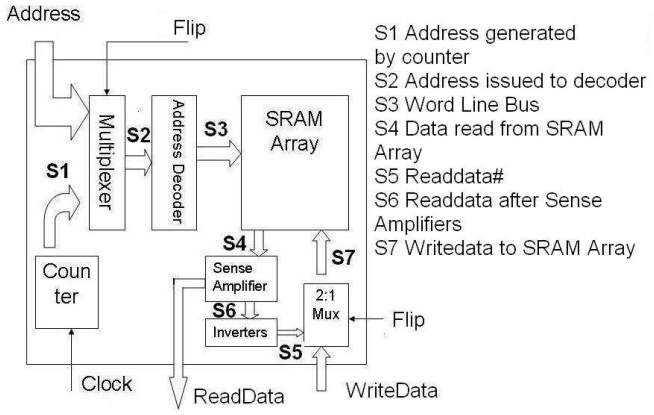

Fig. 7. Hardware implementation showing the additional hardware and control signals needed in existing SRAM arrays for periodic cell flipping

on PMOS transistors). Hence, maximum savings are obtained during prolonged stand-by mode of operation. However, if the processor does not enter stand-by mode exactly at the time desired (say every 10^5 seconds), this flipping mechanism can be performed as soon as the processor enters stand-by the next time. Further, the exact time interval of flipping can also be adjusted based on the feasibility of implementation. The operating system can be scheduled to perform this task on an everyday (every 0.864×10^5 seconds) basis rather than every 10^5 seconds along with other periodic tasks that run everyday. (Simulations showed that flipping the cells every $0.864 \times 10^5 s$ instead of $10^5 s$ gave only $0.2mV$ improvement in SNM which implies that the cell flipping mechanism is almost insensitive to small changes in the flipping interval.)

B. Read and write mechanism modification for flipped SRAM cells

Modifications to the existing read and write mechanism are needed since the data present inside the cache is in its inverted state on alternate days. There exist two approaches to ensure that the data is read and written correctly namely software and hardware approaches.

Software approach: In the software solution, when the processor reads the data on alternate days (days when the contents of the SRAM are flipped), it must flip the contents after it is read. Hence, every read instruction that fetches data from the cache needs to be accompanied by a succeeding INVERT instruction to invert the contents of the read-data bus and interpret it correctly (on alternate days). Similarly, when the data is being written into the SRAM blocks, on alternate days, the inverted data needs to be written. Hence every write instruction needs to be preceded with an INVERT instruction.

Maintaining zero hardware overhead in the software approach ensures that the read and write access times are unaffected. This technique can be implemented in existing systems by modifying the compiler. However, this method requires the insertion of an additional instruction before every write and after every read instruction. The savings in hardware delays can potentially be offset by the additional time it takes to execute these INVERT instructions. Further, in the case of direct data transfer from, say, the L3 to the L2 cache, the data that is transferred to the cache may be incorrect (if true data is written on alternate days instead of inverted data on alternate days) in the absence of efficient synchronization between the cache data transfer controller and the processor.

Hardware approach: In the hardware solution, the SRAM array is equipped with additional circuitry to ensure that the correct data comes out of and goes into the data-bus. This is achieved with the use of additional hardware and control signals as shown in Fig. 8. The read-data after the

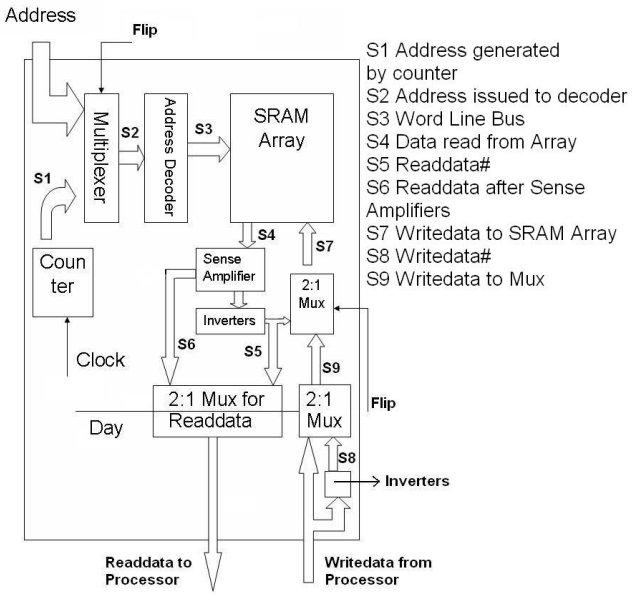

Fig. 8. Hardware implementation to ensure correct data read and write in cell flipping caches

sense amplifiers (S6) is inverted to get read-data# (S5). If the hardware approach shown in Fig. 7 is used for flipping the contents, then this signal is already available. The read-data (S6) and read-data# (S5) signals are then multiplexed using a control signal (Day) which indicates the current state of data (true or inverter) in the SRAM arrays. The final data that comes to the processor is the true data irrespective of the day or state of the cache. Similarly, for writing data into the cache, the processor always sends the true data on the write data bus. Internally this data is inverted, and write-data and write-data# (S8) are both fed to a multiplexer which is similar to the read-mux (controlled by the Day signal). The output of the write-mux (S9) is multiplexed with read-data# (S5) so as to ensure that inverted data is written during cache data flipping and processor data is written during normal course of operation.

This scheme can be adopted for any type of cache and does not require compiler modification to read and write data. The data that comes out of the cache and goes into the cache is always the true data and hence inter-cache data transfer overriding the processor is easily possible. However, the presence of a multiplexer and an inverter on the read and write critical paths affects the access time for read and write.

This may be significant for small cache blocks which are close to the processor.

It must be noted that the above methodology does not take into account the activity factor in the caches and the intrinsic healing effect due to flipping of data during processor writes or replacements. While, generating a model that reflects the operation of the processor and cache blocks, over a period of time as large as three years is seemingly impractical, it can still be argued that the above model rather pessimistically estimates the impact of NBTI on caches (8-9% degradation after 3 years) by not considering the internal cell flipping during normal process operation. Secondly, it must also be noted that intrinsic processor writes may also affect the external cell flipping recovery process causing non-uniform stress and relaxation phases on the PMOS devices in the SRAM cell. Nevertheless, it can be argued that, although caches may be written into frequently, since majority of the data stored in the caches is either 1 or 0, not every bit in every block of the cache is flipped during processor writes/replacements. Further, if certain cache blocks are written into extremely frequently (say at the rate of every 10-100 cycles), the impact of NBTI on these is almost zero, since there is no chance for interface trap build-up, thereby requiring no recovery measures. Hence, the above scheme provides a good performance metric as a baseline measure of the performance recovery obtainable using cache flipping mechanism. The rate of flipping can also be varied in accordance with the activity of the cache blocks. (L1 caches can flip at a much faster rate as compared to say L2 or L3 caches).

VI. Conclusions

NBTI is one of the major concerns of reliability in technologies below 130nm and causes the threshold voltage of PMOS transistors to degrade by 10% after $10^6 s$. This can lead to significant worsening of temporal performance in digital CMOS circuits, especially the SNM of SRAM cells. It has been shown that the SNM degrades by about 8% after 10^8 seconds ($\approx$ 3 years) on 100nm and 70nm cells and can cause read stability issues. A novel technique of cell flipping has been proposed which can recover up to 30% of the noise margin degradation caused due to NBTI. Software and hardware approaches for implementing this technique in data caches have also been discussed.

References

[1] D. K. Schroder and J. F. Babcock. "Negative bias temperature instability: road to cross in deep sub-micron silicon semiconductor manufacturing". *"Journal of Applied Physics"*, 94:1–18, 2003.

[2] V. Reddy, A. T. Krishnan, A. Marshall, J. Rodriguez, S. Natarajan, T. Rost, and S. Krishnan. "Impact of negative bias temperature instability on digital circuit reliability". In *International Reliability Physics Symposium*, pages 248–253, 2002.

[3] S. Mahapatra, P. B. Kumar, and M. A. Alam. "Investigation and modeling of interface and bulk trap generation during negative bias temperature instability of p-MOSFETs". In *IEEE Transactions on Electronic Devices*, pages 1371–1379, 2004.

[4] A. T. Krishnan, V. Reddy, S. Chakravarthi, J. Rodriguez, S. John, and S. Krishnan. "NBTI impact on transistor and circuit: models, mechanisms and scaling effects". In *IEEE International Electronic Devices Meeting*, pages 14.5.1–14.5.4, 2003.

[5] V. Reddy, J. Carulli, A. Krishnan, W. Bosch, and B. Burgess. "Impact of negative bias temperature instability on product parametric drift". In *International Test Conference*, pages 148–155, 2004.

[6] J. G. Massey. "NBTI: what we know and what we need to know - a tutorial addressing the current understanding and challenges for the future". In *IEEE International Integrated Reliability Workshop Final Report*, pages 199–211, 2004.

[7] A. S. Goda and G. Kapila. "Design for degradation: CAD tools for managing transistor degradation mechanisms". In *International Symposium for Quality Electronic Design*, pages 416–420, 2005.

[8] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy. "Impact of NBTI on the temporal performance degradation of digital circuits". *IEEE Electron Device Letters*, 26:560–562, 2003.

[9] G. Chen, M. F. Li, C. H. Ang, J. Z. Zheng, and D. L. Kwong. "Dynamic NBTI of p-MOS transistors and its impact on MOSFET scaling". In *IEEE Electron Device Letters*, pages 734–736, 2002.

[10] "Device Group at UC Berkeley". Berkeley Predictive Technology Model, 2002. Available at `http://www-device.eecs.berkeley.edu/~ptm/`".

[11] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. "New paradigm of predictive MOSFET and interconnect modeling for early circuit design". In *Custom Integrated Circuits Conference*, pages 201–204, 2000.

[12] C. H. Liu, M. T. Lee, C-Y Lin, J. Chen, K. Schruefer, J. Brighten, N. Rovedo, T. B. Hook, M. V. Khare, S-F Huang, C. Wann, T-C Chen, and T. H. Ning. "Mechanism and process dependence of negative bias temperature instability (NBTI) for pMOSFETs with ultrathin gate dielectrics". In *IEEE International Electron Devices Meeting*, pages 39.2.1–39.2.4, 2001.

[13] B. Zhu, J. S. Suehle, Y. Chen, and J. B. Bernstein. "Negative bias temperature instability of deep sub-micron p-MOSFETs under pulsed bias stress". In *IEEE International Integrated Reliability Workshop Final Report*, pages 125–129, 2002.

[14] M. Denais, V. Huard, C. Parthasarathy, G. Ribes, F. Perrier, N. Revil, and A. Bravaix. "Oxide field dependence of interface trap generation during negative bias temperature instability in PMOS". In *IEEE Integrated Reliability Workshop Final Report*, pages 109–112, 2004.

[15] Y. F. Chen, M. H. Lin, C. H. Chou, W. C. Chang, S. C. Huang, Y. J. Chang, K. Y. Fu, M. T. Lee, C. H. Liu, and S. K. Fan. "Negative bias temperature instability (NBTI) in deep sub-micron p +-gate pMOSFETs". In *IEEE International Integrated Reliability Workshop Final Report*, pages 98–101, 2000.

[16] B. Zhu, J. S. Suehle, J. B. Bernstein, and Y. Chen. "Mechanism of dynamic NBTI of pMOSFETs". In *IEEE International Integrated Reliability Workshop Final Report*, pages 113–117, 2004.

[17] M. A. Alam. "A critical examination of the mechanics of dynamic NBTI for PMOSFETs". In *IEEE International Electronic Devices Meeting*, pages 14.4.1–14.4.4, 2003.

[18] S. Mahapatra, P. B. Kumar, T. R. Dalei, D. Sana, and M. A. Alam. "Mechanism of negative bias temperature instability in CMOS devices: degradation, recovery and impact of nitrogen". In *IEEE International Electronic Devices Meeting*, pages 105–108, 2004.

[19] H. Usui, M. Kanno, and T. Morikawa. "Time and voltage dependence of degradation and recovery under pulsed negative bias temperature stress". In *IEEE International Reliability Physics Symposium*, pages 610–611, 2003.

[20] K. O. Jeppson and C. M. Svensson. "Negative bias stress of MOS devices at high electric fields and degradation of MNOS devices". *Journal of Applied Physics*, 48:2004–2014, 1977.

[21] M.A.Alam and S.Mohapatra. "A comprehensive model of PMOS NBTI degradation". *Journal of Microelectronics Reliability*, 45:71–81, 2005.

[22] S. Rangan, N. Mielke, and E. C. C. Yeh. "Universal recovery behavior of negative bias temperature instability [PMOSFETs]". In *IEEE International Electron Devices Meeting*, pages 14.3.1–14.3.4, 2003.

[23] M. Ershov, R. Lindley, S. Saxena, A. Shibkov, S. Minehane, J. Babcock, S. Winters, H. Karbasi, T. Yamashita, P. Clifton, and M. Redford. "Transient effects and characterization methodology of negative bias temperature instability in pMOS transistors". In *IEEE International Reliability Physics Symposium*, pages 606–607, 2003.

[24] Y. Taur and T. Ning. *Fundamentals of modern VLSI devices*. Cambridge University Press, Cambridge, United Kingdom, 2002.