iRMW : A Low-Cost Technique to Reduce NBTI-Dependent Parametric Failures in L1 Data Caches

Shrikanth Ganapathy* Ramon Canal* Antonio González* Antonio Rubio§
*Department d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona, Spain
§Department d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Barcelona, Spain
Email: *{sg, rcanal, antonio}@ac.upc.edu §antonio.rubio@upc.edu

***Abstract*—Negative bias temperature instability (NBTI) is a major cause of concern for chip designers because of its inherent ability to drastically reduce silicon reliability over the lifetime of the processor. Coupled with statistical variations of process parameters, it can potentially render systems dysfunctional in certain scenarios. Data caches suffer the most from such phenomenon because of the unbalanced duty cycle ratio of SRAM cells and maximum intrinsic susceptibility to process variations. In this paper, we propose a novel NBTI-aware technique, *invert*-Read-Modify-Write (iRMW) that can improve the functional yield of the data cache significantly over its lifetime. Using architecture-level benchmarks, we first analyse the impact of activity factor and workload variation on NBTI-induced failures in data caches. iRMW is then used as a means to balance the duty cycle by alternating between recovery and stress cycle upon successive read accesses to the cache line. The highly transient nature of the data stored in L1 data cache aides this process of recovery upon using iRMW. A unique feature of iRMW is its intelligent use of low-leakage & NBTI-tolerant embedded-DRAM cells as an alternative to SRAM-cells for storing important state information. Our experiments conducted using SPEC2006 and PhysicsBench workloads show that on-average the cache failure probability can be reduced by 22%, 33% and 36% after two, four and eight years of processor usage respectively. In addition to being extremely power-frugal, use of eDRAM reduces total area footprint of iRMW tremendously.**

I. Introduction

Continuous advancements in semiconductor manufacturing have been a major component in driving Moore's law, paving the way for ever faster and energy efficient microprocessors. To be able to sustain this pace in the future requires optimizing various design specifications such as silicon real-estate, energy consumption, speed and most importantly, fault-tolerance. Achieving complete fault-tolerance is imperative in the context of modern day designs when considering the impact of manufacturing-induced process variations [1]. Many optimization approaches, like guard-banding and post-silicon tuning can effectively hide the negative impact of process variations albeit being agnostic to time-varying reliability concerns. In such a scenario, several wear-out mechanisms over the lifetime of the chip can cause serious robustness problems resulting in a large number of transient and hard failures causing permanent damage to the underlying silicon. *Negative bias temperature instability* (NBTI) is one such problem that exacerbates failures by affecting the lifetime of PMOS transistors [2].

(Dis)graceful degradation of reliability due to NBTI in logic circuits can be offset to a large extent by carefully up-sizing transistors during the design stage [2]. However, such a technique cannot be applied to on-chip memories due to the stringent constraints set on area and power. On-chip memories designed using SRAM bitcells are the most susceptible to NBTI-induced device mismatch owing to their reduced minimum geometry dimensions. Even at nominal voltages, NBTI can further reduce bitcell noise margins increasing the probability of a bit-flip leading to a failure. Such type of variation-induced failures are called as *parametric failures* and they are responsible for more than 50% yield loss in cache memories [3]. Such failures result in a reduction in cache functional yield (total addressable memory space) across the lifetime affecting key processor-wide metrics such as instructions/cycle (IPC) and performance/watt.

In this paper, we propose a novel technique, iRMW to cope with lifetime performance and reliability degradation in data caches. The scheme particularly improves the lifetime of first-level (L1) caches considering the highly transient nature of the stored data. We validate this notion by executing multiple benchmarks and studying the utilization factor of L1 caches. The observed behaviour is then extrapolated over a longer time period by capturing both NBTI- and workload-dependent reliability reduction to determine the overall degradation in V_{th}. Using obtained metrics and employing statistical circuit-level simulations, we then accurately estimate the failure probability (temporal yield) of the cache. iRMW extends the lifetime of the cache by frequently modifying the data stored in the cache based on access patterns. This reduces the total amount of time the cell is under stress by balancing the duty cycle. Duty cycle is defined as the ratio of time period between storing logic '0' (stress) and logic '1' (recovery). The state information of the changing *bit patterns* are stored in NBTI-hardened embedded-DRAM (eDRAM) cells. eDRAM unlike SRAM technology, is both area-conscious and power-frugal making it an attractive option for storing any kind of meta-data. The primary contributions of the paper can be summarized as follows,

- A novel technique, iRMW that leverages duty-cycle imbalance and r/w access patterns in caches to lower the impact of NBTI-dependent failures.

- A novel logic, Write-back grouping (WBG) to reduce power consumption due to costly write-back operations in iRMW.
- Use of NBTI-tolerant, low-leakage and area-conscious embedded DRAM cells intelligently for storing iRMW specific state information.

The remainder of the paper is organized as follows. In section II, we discuss about NBTI mechanism, parametric failures and L1 cache-specific access patterns. This section serves as a motivation for the proposal. Section III presents the background and related work. In Section IV, iRMW is proposed and in addition, we discuss a new technique, WBG to reduce power consumption in iRMW. Section V presents experimental setup and details the results that include a discussion on Δ V_{th} and cache failure probability reduction using iRMW. Finally, the concluding remarks are presented in section VI.

II. Background and Motivation

A. NBTI Overview

NBTI is a reliability degradation phenomena that is observed at the sub-atomic level. NBTI occurs in two phases, namely *stress* and *recovery*. Periods of *stress* are caused due to generation of interface traps as and when *Si-H* bonds are broken under the influence of high electric fields and elevated temperatures. The dangling bonds created by the separation of hydrogen-terminated trivalent silicon bonds (Si_3-Si-H) create traps at the interface causing hydrogen to diffuse into the gate-oxide. This results in a degradation of the V_{th} of PMOS devices. The annealing process of *recovery* is made possible by the application of V_{dd} to the gate that temporarily inhibits further generation of interface traps. As a result, the dissociated hydrogen bonds return to the interface to join with the broken silicon bonds to partially recover degraded threshold voltage. In this paper, we have modeled the process

TABLE I
Reactive-Diffusion model based threshold voltage shift

	$\Delta\lvert V_{th}\rvert$ under NBTI
Stress	$\sqrt{K_v^2.(t-t_0)^{0.5}+\Delta V_{TH0}+\delta_v}$
Recovery	$(\Delta V_{TH0}-\delta_v).\lfloor 1-\sqrt{\eta(t-t_0)/t}\rfloor$
K_v	$A.t_{ox}.\sqrt{C_{OX}(V_{GS}-V_{TH})}.$ $[1-V_{DS}/\alpha(V_{GS}-V_{TH})].$ $exp(E_{OX}/E_0).exp(E_a/kT)$

of *stress* and *recovery* through the well established *Reaction-Diffusion* (R-D) theory [4], [5], [6]. Table I shows the different parameters influencing the amount threshold voltage shift during stress and recovery phases. It should be observable that the change in the threshold voltage (ΔV_{th}) is directly dependent on the total time under stress ($\sim$ $t^{0.25}$) for a given supply voltage and temperature. By modulating the supply voltage, the overall power-density of the chip can be controlled which in-turn regulates temperature. Temperature has an exponential influence on NBTI *stress* thereby escalating the degradation at high temperatures (heavy-workload or high supply voltage). The degradation exhibited by NBTI is “front-loaded” in nature wherein the rate of degradation reduces with time [7]. By accounting for the supply voltage-speed dependence, the total time under stress can be considerably manipulated lowering the rate of aging at nominal voltages. However, as demonstrated in [7], the returns on dynamic voltage scaling diminish due to the “front-loaded” nature of degradation and it cannot be used to extend the lifetime of the processor. Since the device is only under stress when a negative-bias ($V_{GS}=-V_{dd}|'0'$) is applied, the duty cycle (β) is an important parameter that can influence ΔV_{th} significantly [8]. Figure 1 shows the impact of duty cycle variation on the

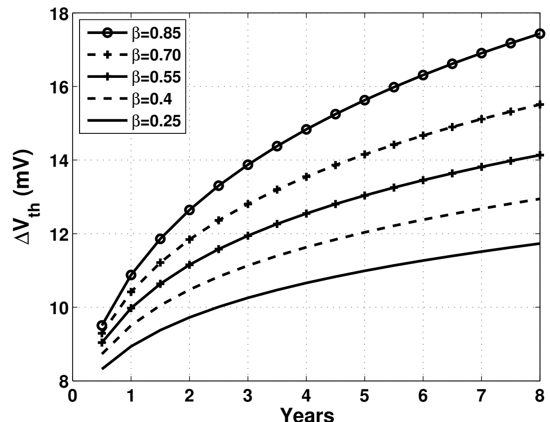

Fig. 1. ΔV_{th} shift in 32nm PMOS device operating at 85°C

ΔV_{th} for a 32nm PTM PMOS device operating at 85°C [9]. It can be observed that the rate of degradation (slope of ΔV_{th}) is higher for a device with larger duty cycle. Higher shifts in ΔV_{th} require higher margin of V_{dd} or temperature tuning to ensure sufficient recovery. Unlike other design specifications, duty cycle cannot be tuned arbitrarily and the effectiveness of duty-cycle modulation needs to be studied together with other metrics such as component utilization factor and input vector probability.

SRAM-based memory cells are composed of a pair of inverters connected in a positive feedback loop wherein each inverter charges the gate of the PMOS or NMOS of the other inverter. Under the influence of NBTI, the PMOS of one inverter will always be under stress. Coupled with variation in the threshold voltage due to process variations, the static noise margin of the cell degrades over time increasing the probability of a failure. Figure 2 shows the measured failure probability for a SRAM cell under varying standard deviations of V_{th}. The estimated failure probabilities consider only static variations of V_{th} and not the time-dependent V_{th} variation (PMOS only) due to NBTI. It was shown in [10], when considering the impact of NBTI also, read failures increase by 3 orders of magnitude for a 20mV increase in standard deviation. Read failure probability is also dependent on the data stored as this influences the amount of V_{th} degradation in the cell. In the context of processor-wide performance penalty, this is extremely important as read accesses to L1 caches lie on the critical path of the processor pipeline. It is possible to lower read failure probability by balancing the input signal

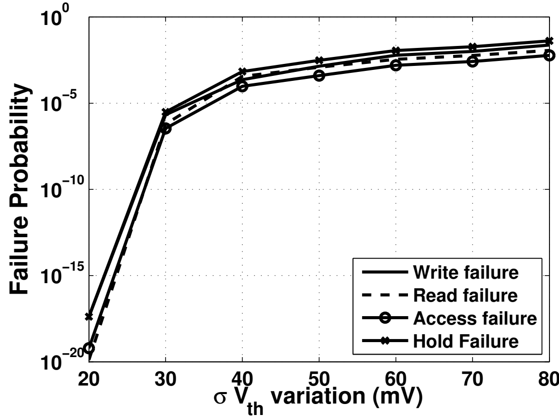

Fig. 2. Parametric cell failure probability for varying σV_{th}

probability. Write failures on the other hand, reduce with time as with increasing V$_{th}$ it becomes easier to write logic '0' to a node storing a '1'. As a result, write margin improves with time lowering write failure probability. Hold failures experience a minimum increase as the effect of PMOS devices in the standby mode itself is very minimal.

B. L1 cache access patterns

The principal motivation behind proposing an NBTI-aware L1 data cache design is the need to exploit the unbalanced duty cycle ratio exhibited both spatially and temporally. Previous studies have highlighted that more than 75% of the time, logic bit value '0' is stored in the cells [11]. By frequently inverting the contents of the cell, it is possible to lower logic bit '0' occupancy to around 50%. Even in this case, the cell is degrading but in a more balanced manner. Periodic flipping in caches can ensure adequate recovery only when the data is stored for a sufficiently long period without being evicted. Upon running several benchmarks (see Section IV for the experimental details), we notice that cache lines in L1 caches are accessed frequently and modified within very short intervals as shown in figure 3. It can be observed that in the worst-case (povray), a cache-line is accessed no later that 300,000 cycles after it was first written. The data points in the graph represent the longest periods between successive writes to any cache line. The references to the cache here are specifically read accesses. The temporal data characteristics exhibited by the benchmarks show that in most cases, the accesses are well spread (linear) across the lifetime of the data before its evicted. In other words, the total period time during which a cache line is 'dead' is minimal when compared to the total time before eviction. If it is possible to modify the data bits frequently during its *live* phase, then it is safe to assume that it is possible to achieve near-perfect duty cycle balance.

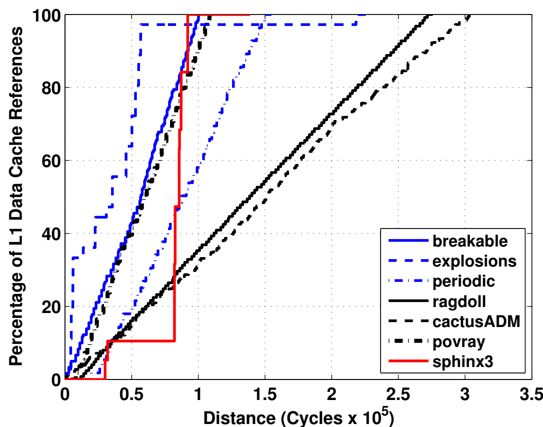

Fig. 3. Percentage L1 data cache references versus elapsed time

III. Related Work

The effect of NBTI-induced parametric failures in SRAM cells was studied and a SW/HW solution based on bit-flipping was proposed in [12]. The solution exploits the unbalanced duty-cycle of SRAM cells by flipping the contents of the cell periodically (wall-clock) thereby ensuring both halves of the SRAM cell degrade symmetrically lowering failure rates. The coarse-grain nature of the mechanism in operating at the memory array level makes it agnostic to data placement and access patterns where certain physical regions (words or lines) of the cache can undergo varying levels of degradation depending upon its access characteristics during its lifetime. In [13], a technique that uses redundancy proactively to lower the impact of NBTI on memory yield is proposed. The technique periodically migrates data from an active array to a spare memory array and puts the active array in recovery (inactive) mode for lowering NBTI effect. However, in a high-activity structure like a L1 data cache where adjacent logical addresses can be stored across multiple physical arrays, maintaining data coherence and correctness during this transition period can become a cumbersome process that requires complex micro-architectural support. Our technique is *self-contained*, in that the changing *bit patterns* are hidden from program execution (micro-architecture) wherein data modifications are always effected after a *write access* and before a *read access*. Aging-aware data cache scheme proposes a micro-architecture scheme to flip the contents of a cache line after a fixed (static) prolonged period of idleness (after which it is assumed that the line will never be accessed - *dead*) [11]. A global counter is used to count the period of idleness and after certain number of clock cycles during which there are no read accesses idle), the line is invalidated and the whole line is flipped (either 0 or 1). However, as we have shown in figure 3, for some applications, subsequent read accesses can be separated by several thousands of cycles and forcefully invalidating a line before its last read access requires bringing the data from the lower levels of the memory when accessed at a later point incurring significant performance losses. In our technique we do not make any assumptions about the access patterns of the underlying application and we do not have to explicitly maintain data coherency as required by the proposal in [11].

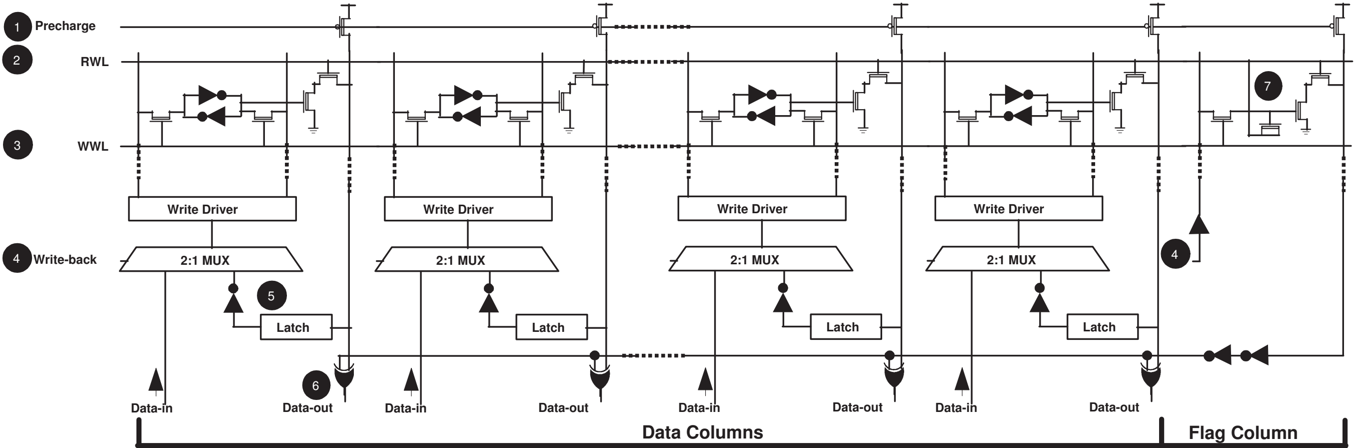

Fig. 4. **iRMW**: invert read-modify-write technique for lowering NBTI-dependent failures

IV. INVERT READ-MODIFY-WRITE

One of the main constraints while designing new cache architectures optimized for low power and reliability is $V_{cc_{min}}$ of the 6T-SRAM cell. It is defined as the minimum voltage that guarantees fault-free operation of the memory. It was shown in [14] that processor-wide $V_{cc_{min}}$ is dependent on the highest $V_{cc_{min}}$ among all cells across all arrays. Under the effects of random variations where failures can be distributed, a single 6T-SRAM cell can potentially affect the operational margin of the processor if no repair mechanism is in place.

In view of such stability issues, an alternate proposal introduced by *Chang et al.* is the 8T-SRAM cell [15]. An 8T cell (as shown in figure 4) is a 6T cell with additional two transistors to provide a new read path without disturbing the internal storage nodes. In addition to providing a multi-ported design, the read and write margins are improved tremendously by decoupling the r/w paths. This lowers the parametric failure probability significantly. Further to doubling the bandwidth for simultaneous access, for technologies below 45nm, 8T cells become smaller compared to their 6T counterparts [16]. However, in bit-interleaved architectures, cells of unaccessed adjacent columns are also triggered (wordline high) making them especially susceptible to read failures. This is caused due to the single-ended bitline architecture where a large noise developed on the bitlines can upset the value stored in the cell during a read. This problem is commonly referred to as the *half-select* issue. It can be avoided by designing arrays with hierarchical bitlines or employing a read-modify-write scheme where a read access is always followed by a write data back operation [16]. iRMW (shown in figure 4) is one such approach that in addition to coping with dynamic variations of NBTI, can mitigate the *half-select* problem.

1) Proposed iRMW logic: When a particular cache line receives a write access request, the write wordline (WWL) drivers rise WWL to signal a write operation. The data inputs to the write operation are controlled by a multiplexer that selects between a *write* or *write-back* operation. For the *write* operation, the write drivers are loaded values from *Data-in*.

Operation ⟶	Flag	Cell Value	Data-Out
Write	0	0	0
Write	0	1	1
Write-back	1	0	1
Write-back	1	1	0

TABLE II
XOR LOGIC DURING READ FOR INVERTING INPUT/OUTPUT BASED ON WRITE/WRITE-BACK OPERATION

This is the data received from the system-bus. For the *write-back* operation, the values are loaded from the latches. Additionally, during *write* or *write-back* request, the multiplexer-selection-input (logic “0” for write and logic “1” for write-back) value is written into a eDRAM memory cell (flag value). This is a three-transistor one-diode DRAM (3T1D) cell that supports non-destructive reads with access speeds on par with regular SRAM cells. Therefore, unlike regular DRAM cells, a refresh is not needed after every read access (power reduction). Also, the NMOS-only design makes it completely tolerant to NBTI-induced variations. A read operation is initiated by precharging the bitlines to V_{cc} and driving read wordline (RWL) high. Depending on the value stored in the cell, the bitlines either charge or discharge to fire the sense-amplifier. To avoid the *half-select* issue, during every read access, the data is simultaneously stored in a latch (one per column) for safeguarding. At the end of the read access, a write-back operation is initiated which involves inverting the contents of the latch and writing it back into the cell. The value read during the read access is XOR’ed with the value of the flag bit that represents the state of the latest write request (either a write or a write-back). The first time a cell is accessed (read) after it is written, the contents have not been modified yet as the last write request was only a write and not a write-back. At the end of the read access, the write-back is initiated where the contents of the latch are inverted and written into the cell. Simultaneously, the flag is also updated to signal the last write request was a write-back. The logic of this operation is shown in table II.

2) Write-back grouping: As a write-back operation is being serviced, the cell cannot be read in parallel. This is a major drawback for L1 caches especially as it reduces the available bandwidth penalizing the performance. The need for a compulsory read before write contributes additionally to power dissipation. We propose a new technique called *Write-back grouping* (WBG) that is a modified version of the write-grouping proposal presented in [17] to address these concerns. The address of the last-accessed (write) cache set and its associated tags are stored in a small buffer inside the cache controller called the *Tag-buffer*. Additionally every set also has a counter that counts till a predefined *threshold* which is set to 0 after every write. Upon the first read request (after write), the contents of the *tag-buffer* are checked to determine if the address present is the same as the requested one. If there is a match and the counter is 0, then a write-data back operation is initiated. At the end of this operation the counter is also incremented. As long as there are continuous hits to the same address, the counter is incremented by 1 till it reaches the threshold value. During this phase, no write-backs are performed. Once the threshold value is reached, the write-back operation is performed again and the flag is modified to signal a write operation indicating that the contents of the cell have been restored to its original value. In the event of a read miss, the counter is set to 0 and with the eviction of data, the contents of the *tag-buffer* are modified to reflect the new index and tags. The technique exploits *set access locality* exhibited by workloads where a large portion of the accesses are to the same set ($\sim$ 30%-35%). We modified the iRMW logic to support WBG and set the threshold value to 50. Therefore, we perform the write-back operation to the same address only once in every 50 read requests. The choice of threshold values is prompted by the distance (in-cycles) that is small enough to avoid the "front-loading" effect yet large enough considering set access locality to ensure significant power savings by avoiding costly write-back operations. Figure 5 shows the impact of r/w access patterns on threshold voltage degradation due to NBTI. For the baseline design, degradation is dependent on the value being written and the time for which it is stored. iRMW on the other hand conveniently relies on read requests (minimum 2X more than writes) to forcefully push the cell into recovery. Further, the shift between stress and recovery occurs at a more fine-grain level compared to the baseline where the effects of *front-loading* are more pronounced.

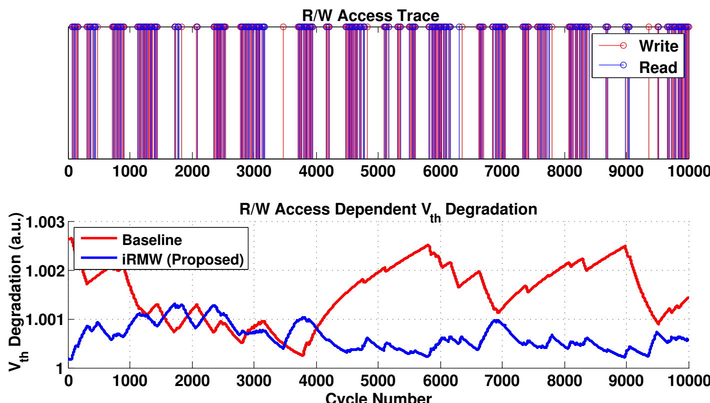

Fig. 5. R/W access pattern dependent V_{th} degradation for the benchmark **450.Soplex** for a period of 10000 cycles chosen arbitrarily

3) 3T1D eDRAM for storing state information: Another concern in iRMW is the inability of the 3T1D cell to store the contents of the flag perpetually. Therefore, there is a definite chance that contents of the flag will be lost forever. We alleviate this problem by running exhaustive simulations to determine the maximum distance (in cycles) between two consecutive read accesses. First, performing the write-back operation updates the flag frequently ensuring the retention of the data for a sufficient period. Next, we define the counter threshold value based on the maximum number of requests within an interval defined by the retention time of the cell. We designed a cell that has a guaranteed (considering process variations) minimum retention time of 20μs and it can support SRAM-like access speeds for the first 6μs after the contents are written. This translates to roughly 20000 processor cycles (@ 3.3GHz) during which the 3T1D cell can be accessed with SRAM-like speeds. As shown in figure 3, the worst-case distance between two consecutive read accesses is no more than a few thousand cycles which is very much within the retention time and speed-assurance period of the 3T1D cell. It should be noted that the eDRAM cell used in this proposal is logic compatible meaning it can be integrated on-chip without extra manufacturing process steps and several previous studies have discussed in detail the feasibility of a L1 cache design using such eDRAM cells [18], [19].

V. Experimental Evaluation

A. Simulation Setup

1) Architecture-level simulations: Our architecture-level analysis is performed using DARCO simulation infrastructure that is targeted more towards evaluating HW/SW co-designed virtual machines [20]. x86 binary instructions are executed on a PowerPC-like RISC host architecture. It is assumed that the host-architecture consists of a 128-bit wide SIMD accelerator that is composed of two 64-bit wide lanes. The workload consists of a subset of applications from SPEC2006 and PhysicsBench suites [21]. The benchmarks are instrumented using PIN to determine the most frequently executed routines and these are replayed continuously. We fast-forward 100 million instructions for warming up the cache and then record the cache traces for the next 10 million accesses (not instructions). The L1 data cache is 64KB, 4-way set associative with a 64-byte line size. For performance reasons, it is assumed that the accesses are coupled with a 1-cycle hit latency. LRU is the default replacement policy.

2) Circuit-level simulations: To capture the impact of NBTI through architecture-level simulations, we employ the approximations in [22] to estimate the V_{th} degradation after two, four and eight years of processor usage. Without loss of generality, it is assumed that 50% of the time, logic '0' is written into the cell. The duty cycle is different as it depends on the occupancy of the value dependent on access patterns. Then we derive the NBTI-dependent threshold degradation based on average

occupancy rate, clock cycle and duty cycle using R-D model. The total degradation in threshold voltage is then modeled using the equation shown below.

$$\sigma V_{th} = \sqrt{\sigma V_{th_{PV}}^2 + V_{th_{NBTI}}^2} \quad (1)$$

With the obtained threshold deviation, we then run exhaustive DC and transient spice-level simulations to estimate the failure probability of the 8T-SRAM cell. Input parameter values used for estimating failure probabilities are obtained from [23]. The cell is designed using 32nm PTM technology model operating at 0.9V [9]. The run-time ambient temperature is 70°C.

B. Results

1) V_{th} Degradation: We study the impact of NBTI on V_{th} degradation after a period of two, four and eight years of processor usage. As we do not simulate the actual data in the cache and provide random data inputs during writes, we run the test suites repeatedly inside our simulation framework for a large number of times until the measured output metrics converge within 5% of the mean values. Figure 6 shows the percentage reduction in V_{th} degradation for iRMW when compared to the baseline architecture. Over here, the baseline architecture does not employ any mechanisms to combat NBTI. For the sake of brevity, we show only the degradation after four years (median lifetime) as the relative reduction in degradation between iRMW and baseline is found to be consistent across the lifetime. It can be seen that compared

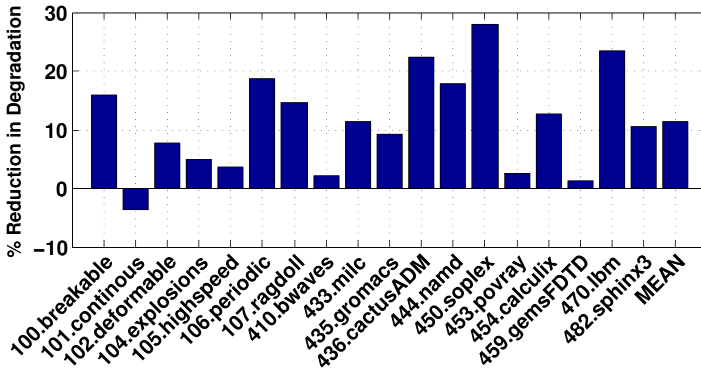

Fig. 6. Reduction in V_{th} degradation due to NBTI for iRMW compared to baseline after four years of processor usage

to the baseline, the average reduction in degradation is of the order of 10%. When comparing the results in conjunction with the analysis presented in section IIa and IIb, it should be clear that even a minimal reduction in V_{th} degradation should yield higher benefits when considering system-wide metrics. In general, iRMW balances the duty-cycle tremendously by frequently shifting between stress and recovery. By choosing an optimum threshold value, the period of stress is only large enough that it does not warrant a write-back operation and yet small enough to ensure there is maximum recovery during the phase after the next write-back. Only for one benchmark (*101.continuous.*) does iRMW perform worse than the baseline. Compared to the other benchmarks, this particular benchmark was characterized by large periods of idleness that were followed by short bursts of large number of accesses. While the degradation during the short period of accesses is not much as we toggle between stress and recovery, assuming 50% of the idle times the memory is under stress, then it would require sufficiently longer period for recovery. The much needed period for recovery was seldom available and bursts of accesses exacerbated the *front-loading* effect ensuring there was constant degradation when toggling.

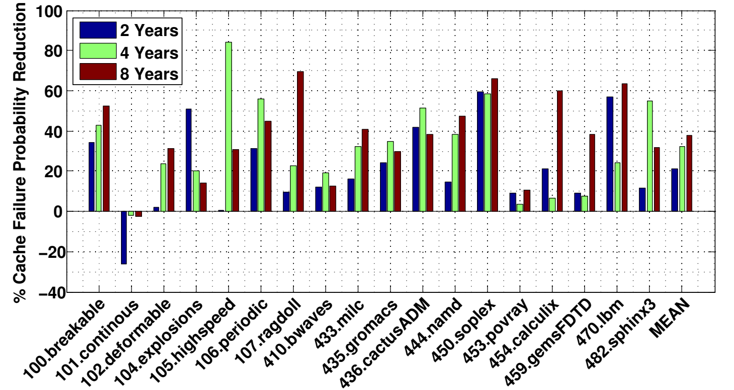

Fig. 7. Reduction in cache failure probability for iRMW compared to baseline after two, four and eight years of processor usage

2) Cache failure probability: Figure 7 shows the reduction in overall cache failure probability of iRMW when compared to the baseline architecture. We normalize the cache failure probabilities to the baseline value obtained after two, four and eight years of usage. On average the failure probability is reduced by 22%, 33% and 36% after two, four and eight years of processor usage. In line with the results presented in figure 6, the cache failure probability increases by as much as 20% for *101.continuous.* Also, it should be observable that the reduction in failure probability is inconsistent for other benchmarks across the lifetime and with comparison to the degradation in V_{th} presented in figure 6. This can be attributed to the methodology we adopt to determine the failure probability of the 8T-SRAM cell. From a simulation standpoint, it is very expensive to simulate events with extremely low failure probabilities (after two/four years usage). Relying on importance sampling techniques for simplification cannot guarantee consistent results across simulation runs for the same input due to the random nature of simulation methodology. To reduce the variance in the results, again we run the tests repetitively and report the mean of the outputs metrics. Adopting such heuristics, we estimated our results to be within a differential of 3 to 8% of the actual estimates.

3) Overheads: One flag column is shared among 32 columns (read-width). If the block-size|read-width is larger, the area overhead is significantly reduced. The embedded DRAM cell used as a flag for storing state information is 40% smaller when compared to a regular 6T-SRAM cell in 45nm. As per reported estimates in [18], in 32nm technology node the area of the 3T1D cells is $0.23\mu m^2$. In the same technology, the average cell area of the 8T-SRAM is $0.34\mu m^2$ making the 3T1D 33% smaller compared to the 8T. As the

cell is rewritten frequently (much before maximum retention time), the need to refresh is completely avoided lowering the dynamic power overhead. The leakage power on-average is 4X lesser compared to a regular 6T-SRAM [18]. The total area required by the Tag-Buffer is approximately 150 bits considering a 48-bit virtual address [17]. The largest area-overhead is incurred due to the counter that maintains the number of read requests received. The counter-area is dependent on the maximum *threshold-value* that can be accommodated. Over here, *threshold-value* is a good metric to trade-off reliability for power and area of the counter during design stage. The overhead due to counter is expected to be within 5%-8% of cache area.

VI. Conclusions

In this paper, we investigate the impact of NBTI-dependent parametric failures in L1 data caches. We categorically show that the imbalance in duty-cycle and L1 data cache-specific access patterns can be exploited to improve the lifetime of such caches. We then propose a new technique, iRMW that modifies the bit patterns periodically to toggle between stress and recovery phases to lower V_{th} degradation. As the modification requires an extra operation (write-back) that incurs additional power overhead, a new supporting logic, WBG is also proposed. Write-backs are initiated consistently (neither frequently or sparsely) thereby alternating between stress and recovery for larger time-periods. This helps lower the power consumption significantly. Our results using architectural level benchmarks shows that iRMW can reduce overall cache failure probability by 22%, 33% and 36% after two, four and eight years of processor usage respectively.

VII. Acknowledgments

This work has been partially supported by the Spanish Ministry of Education and Science under grant TIN2010-18368 and TEC2008-01856, the Generalitat of Catalunya under grant 2009SGR1250 and Intel Corporation. We also would like to thank Rakesh Kumar (Department d'Arquitectura de Computadors, Universitat Politècnica de Catalunya) for his contributions towards setting up the DARCO simulation infrastructure.

References

[1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of the Design Automation Conference*, 2003.

[2] K. Kang, S. Gangwal, S. P. Park, and K. Roy, "Nbti induced performance degradation in logic and memory circuits: how effectively can we approach a reliability solution?" in *in Proceedings of the Asia and South Pacific Design Automation Conference*, 2008.

[3] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou, "Yield-aware cache architectures," in *Proceedings of the International Symposium on Microarchitecture*.

[4] S. Mahapatra, P. Bharath Kumar, and M. Alam, "Investigation and modeling of interface and bulk trap generation during negative bias temperature instability of p-mosfets," *IEEE Transactions on Electron Devices*, 2004.

[5] "A comprehensive model for {PMOS} {NBTI} degradation: Recent progress," *Microelectronics Reliability*.

[6] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pmos nbti effect for robust nanometer design," in *Proceedings of the Design Automation Conference*, 2006.

[7] T.-B. Chan, J. Sartori, P. Gupta, and R. Kumar, "On the efficacy of nbti mitigation techniques," in *Proceedings of the Design Automation Test in Europe Conference Exhibition*, 2011.

[8] A. Ricketts, J. Singh, K. Ramakrishnan, N. Vijaykrishnan, and D. Pradhan, "Investigating the impact of nbti on different power saving cache strategies," in *Proceedings of the Design, Automation Test in Europe Conference Exhibition*, 2010.

[9] "Predictive Technology Models, http://www.eas.asu.edu/ ptm."

[10] K. Kang, H. Kufluoglu, K. Roy, and M. Alam, "Impact of negative-bias temperature instability in nanoscale sram array: Modeling and analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2007.

[11] S. Wang, G. Duan, C. Zheng, and T. Jin, "Combating nbti-induced aging in data caches," in *Proceedings of the International Conference on Great Lakes Symposium on VLSI*, ser. GLSVLSI '13, 2013.

[12] S. Kumar, C. Kim, and S. Sapatnekar, "Impact of nbti on sram read stability and design for reliability," in *Quality Electronic Design, 2006. ISQED '06. 7th International Symposium on*, March 2006, pp. 6 pp.–218.

[13] J. Shin, V. Zyuban, P. Bose, and T. M. Pinkston, "A proactive wearout recovery approach for exploiting microarchitectural redundancy to extend cache sram lifetime," in *Proceedings of the International Symposium on Computer Architecture*.

[14] S. Ganapathy, R. Canal, A. Gonzalez, and A. Rubio, "Effectiveness of hybrid recovery techniques on parametric failures," in *Proceedings of the International Symposium on Quality Electronic Design*, 2013.

[15] L. Chang, D. Fried, J. Hergenrother, J. Sleight, R. Dennard, R. Montoye, L. Sekaric, S. McNab, A. Topol, C. Adams, K. Guarini, and W. Haensch, "Stable sram cell design for the 32 nm node and beyond," in *in Proceedings of the VLSI Technology (Digest of Technical Papers)*, 2005.

[16] Y. Morita, H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "An area-conscious low-voltage-oriented 8t-sram design under dvs environment," in *Proceedings of the Symposium on VLSI Circuits*, 2007.

[17] M. Farahani and A. Baniasadi, "Performance and power solutions for caches using 8t sram cells," in *Proceedings of the International Symposium on Microarchitecture Workshops (MICROW)*, 2012.

[18] X. Liang, R. Canal, G.-Y. Wei, and D. Brooks, "Replacing 6t srams with 3t1d drams in the l1 data cache to combat process variability," *IEEE Micro*, 2008.

[19] A. Teman, P. Meinerzhagen, R. Giterman, A. Fish, and A. Burg, "Replica technique for adaptive refresh timing of gain-cell-embedded dram," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2014.

[20] D. Pavlou *et al.*, "Darco: Infrastructure for research on hw/sw codesigned virtual machines," in *AMAS-BT'11*.

[21] J. L. Henning, "Spec cpu2006 benchmark descriptions," *SIGARCH Comput. Archit. News*.

[22] T. Siddiqua, S. Gurumurthi, and M. Stan, "Modeling and analyzing nbti in the presence of process variation," in *Proceedings of the International Symposium on Quality Electronic Design*, 2011.

[23] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanoscaled cmos," *Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, 2005.