The original of this paper can be found in http://dx.doi.org/10.1109/TVLSI.2014.2355873

Adaptive Proactive Reconfiguration: A Technique for Process Variability and Aging Aware SRAM Cache Design

Peyman Pouyan, Esteve Amat, Antonio Rubio

Abstract — **Nano-scale circuits are subject to a wide range of new limiting phenomena making essential to investigate new design strategies at the circuit and architecture level in order to improve its performance and reliability. Proactive reconfiguration is an emerging technique oriented to extend the system lifetime of memories affected by aging. In this paper, we present a new approach for SRAM memory design that extends the cache lifetime when considering process variation and aging in the memory cells by using an adaptive strategy. To track the aging in the SRAM cells we propose an on-chip monitoring technique. Our results show the technique as a feasible way to extend the cache lifetime up to 5X.**

Index Terms—Adaptive proactive reconfiguration, Aging sensor, Process variation, Reliability, SRAM

I. Introduction

PROCESS variation and aging caused by Bias Temperature Instability are two key reliability concerns [1] in modern technologies, which relevance has been intensified in the deep nano-scale integration levels. SRAM is one of the main sections in integrated circuits susceptible to such type of deviations due to its extreme sensitivity to process variations [2]. Moreover, memories store information for a long period of time; this causes a long electrical stress period that could cause failure [3].

Cache memories are usually designed with several spare columns/rows in order to substitute the failing ones for yield improvement purposes [4]. By using spare parts and built-in circuit test procedures (after manufacturing or in-field), a reconfiguring scheme allows the substitution of defective parts by spare ones, resulting in a self-repairing fault-tolerant. This principle is called reactive configuration [5]. In this strategy the reserved spares are not operative until a fail is detected. An alternative is that instead of saving all the spare units up to time of failure, they could also take part in the normal operation of the system in a technique called proactive reconfiguration [5, 6] sharing the aging phenomena among all the devices, causing a lifetime extension of the memory.

This paper presents an adaptive variation and aging-aware dynamic approach and is organized as follows: Section II describes process variation and BTI aging. Section III introduces the concept of proactive reconfiguration. Section IV describes the basis and motivation of our adaptive technique and in Section V we evaluate our methodology. Section VI analyzes our proposed architecture with our monitoring circuits and test procedure.

"This work was supported by the European TRAMS project (FP7 248789), and the Spanish MINECO (JCI-2010-07083) and TEC2008-01856 ".
The Authors are with Department of Electronic, Polytechnic University of Catalonia (UPC), Barcelona, Spain, (e-mail: peyman.pouyan@upc.edu, esteve.amat@upc.edu, antonio.rubio@upc.edu).

Finally, Section VII highlights the results obtained throughout the paper.

II. Considerations About Variability and BTI Aging

A. Process variability

Process variability is a reliability related issue, which has gained a significant importance at nano-scale level circuit design. It is mainly caused by advanced manufacturing process and is divided into two major types, i.e. systematic and random deviations [1]. These variations along with the wear-out that occurs during the system lifetime (aging) reduce the system's performance and lifetime.

B. BTI Aging

There are several degradation mechanisms in nano-scale devices, which are well reported in literature, i.e. Bias Temperature Instability (BTI) and Channel Hot-Carrier (CHC) injection [7]. In this work, we only consider BTI aging, since it is the main aging mechanism that leads to relevant V_T-shift and could result in a SRAM failure [2, 3] by degrading the SNM value. Several models have been described to predict the impact of BTI aging on V_T-shift in devices [8, 9]. Since our work implies an analysis during a long period of time, we use a simplified piece-wise linear aging model [10]. Moreover, we have assumed that all SRAM cells are based on high-k transistors as correspond to modern technologies, and then both N/PMOSFETs are subjected to BTI aging.

It is known that in the stress phase, the device suffers some V_T-shift due to specific physical mechanisms affecting materials [3, 11]. When the device is released from stress, a second behavior in BTI aging is observed named the recovery phase [9]. This may mitigate some part of V_T-shift and as a consequence extends the device lifetime. We have modeled this wear-out recovery in respect to different possible technologies by using a parameter named recovery factor (Rf) that models the recovery capability of the stressed devices after a large enough relaxing phase. This Rf parameter corresponds to the proportional amount, percentage,

of the V_T-shift value that can be mitigated in a device after a large enough recovery phase. This is due to recovery properties of the BTI aging mechanism and it is technology dependent [7,9]. The maximum possible wear-out recovery is obtained after a recovery time (T_R) of 10^4 seconds [9] and because our relaxing time will be much larger, we assume that the complete recovery can be reached in each phase. The aging model for coming technologies can also further be imported and considered in our design proposal.

III. Proactive Reconfiguration Technique

Proactive reconfiguration techniques can be implemented in different ways, depending on the redundancy granularity. The hardware granularity level selected in this work is the column level [12]. We evaluate the different proactive approaches by employing an analytical Matlab engine [13] and compare the different methodologies. The first approach that we analyze is the approach pointed out by IBM in [5], which corresponds to a basic proactive technique. The IBM proposal is based on a homogeneous round robin strategy between memory columns, where all the columns go to recovery mode homogenously and sequentially, one by one in a rotating schedule [5] without considering inherent process variability and differential aging between the columns. Fig. 1 presents graphically the time-varying aging evolution and the lifetime behavior for two approaches, reactive (non-proactive) and IBM's in an example case. We assume a system example composed by four working columns, one spare and a recovery parameter of Rf=30%. In our example, a set of arbitrary fresh V_T values is stated for the worst V_T (the cell with lowest SNM) in each column. We observe that for the proactive reconfiguration case the aging slope of the evolution transistors' V_T values changes causing an improvement of the system lifetime about 1.8X (150 months in front of 84). In both approaches the system fails when the weakest column arrives to the maximum acceptable value (400mV in the example). Note that, the IBM approach uses equal recovery periods, and therefore it cannot mitigate the relevant time zero process variations of the transistors in SRAM memories. A variation-aware proactive technique, presented in next sections, solves this by taking into account the device variability, as well, and further enhancing the system lifetime.

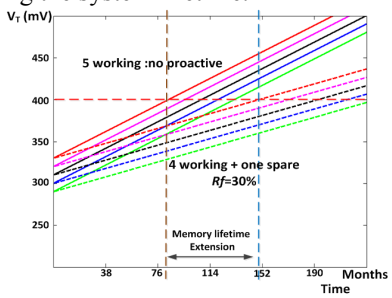

Figure 1. V_T aging slopes of columns in a proactive (dashed lines) and a non-proactive approach (lines). Almost 2X lifetime extension is obtained.

IV. Basis of our Variability Aware Proactive Technique: A Static Non-Homogeneous Approach

The proactive usage of existing available redundant units in the system gives the opportunity to each system element to go to recovery mode some time during its lifetime, since there exist some spares units in the system available to substitute the functional ones. As a consequence, when the column enters into recovery phase some part of its wear-out is mitigated. By adapting the recovery time of each element in the system in accordance with its time zero process variation, we could optimize the observed aging of the system elements, and as a consequence extend their lifetime. Therefore, the V_T values converge toward a common point meaning the optimal lifetime. It was shown in [10] that the lifetime of an SRAM affected by time-zero process variation and aging can be optimized by assigning a relative working time D_i to each column (i) of the memory (N columns, R spare) given in (1):

$$D_i = \frac{(R \times H) + (N - R - 1) \times V_{T_i} - \sum_{j=1, \neq i}^{N} V_{T_j}}{(N \times H) - \sum_{j=1}^{N} V_{T_j}} \quad (1)$$

where H is the maximum V_T value of a cell before to fail and V_{Ti} the worst case cell threshold voltage for each column. This technique allows to manage the aging of the memory columns in such a way that they all converge to a common point during their lifetime. This would avoid the dependence of the system lifetime to the worst column's lifetime of it. In next section, we apply this technique to a dynamic adaptive basis and compare three configurations, i.e. non-proactive or reactive, IBM proactive and our proactive technique.

V. A Dynamically Adaptive Extension to the Proactive Technique

Our adaptive proactive reconfiguration is an improved version of previous proactive reconfiguration [5] in which its utilization results in a balanced aging distribution and larger lifetime extensions throughout the memory columns. First, we expose the approach flow of our methodology and our recovery time period calculation for each memory column. Next, we show the results of our adaptive proactive approach.

A. *Approach Flow and Dynamic Recovery Calculation*

Our approach is based on a non-homogeneous round robin sequence between all the memory columns that also considers and self-adapts the process variation and BTI wear-out of SRAM cells in a time-varying basis. The utilization of spare units allows us to make a test in the memories to determine the status of memory columns. It also permits to define different recovery times, which can be dynamically adapted to the respective V_T values. It starts with a test that measures the V_T value of each SRAM cell. Then, each monitored column will be characterized by its highest V_T SRAM cell (the weakest cell in the column). These measured values, determine the needed recovery time length (D_i) for each memory column. Fig. 2 depicts the procedure flow of our adaptive technique. After the test, the memory columns will be sorted from minimum to maximum (from the column with SRAM cell of highest V_T to the column with SRAM cell with lowest V_T), and according to V_T values the appropriate recovery periods are calculated.

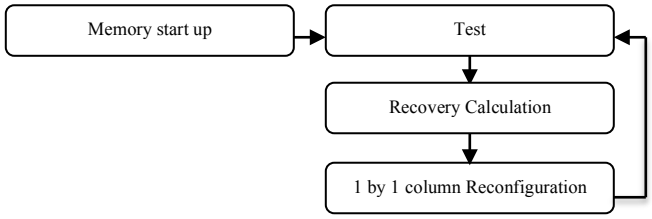

Figure 2. Adaptive proactive approach flow.

Our recovery period calculation approach is based on the range between the weakest and strongest SRAM cells V_T values (min is the value of the minimum (best) V_T column, and max is the value of the maximum (worst) V_T column) in the memory columns. Firstly, we consider a number of V_T ranges in which we want to classify the memory columns among them. Then, we calculate the ΔV_T, which is the difference value between the best and worst column V_T values. Finally, the specific ranges are determined by the mentioned values. The columns are divided between these ranges such that the columns with higher V_T values will have longer recovery times. For instance, we have considered an example in Table I where the number of ranges is equal to 4. Note that the recovery periods are multiples of T_R (10^4 seconds), which is the minimum, needed time for a complete BTI recovery [9].

TABLE I. ROUND ROBIN DYNAMIC RANGES FOR OUR EXAMPLE.

V_T-ranges	Recovery time
$min<V_T<(min)+(1/4\times\Delta V_T)$	$1\times T_R$
$(min)+(1/4\times\Delta V_T)<V_T<(min)+(2/4\times\Delta V_T)$	$2\times T_R$
$(min)+(2/4\times\Delta V_T)<V_T<(min)+(3/4\times\Delta V_T)$	$3\times T_R$
$(min)+(3/4\times\Delta V_T)<V_T$	$4\times T_R$

B. *Single Spare Proactive Reconfiguration Case*

In each reconfiguration step of adaptive proactive reconfiguration, the spare column replaces the working column that goes into recovery mode and the column's data is copied in the spare column. When the memory column becomes active the copied data is written back before the next column reconfiguration step. Fig. 3 presents our adaptive proactive technique among the 5 memory columns. It shows that our approach with an Rf=30% extends further the memory columns lifetime in presence of process variability and BTI aging times. The obtained value is a 25% better than the IBM approach (dashed lines of Fig. 1), 190 months in front of 152.

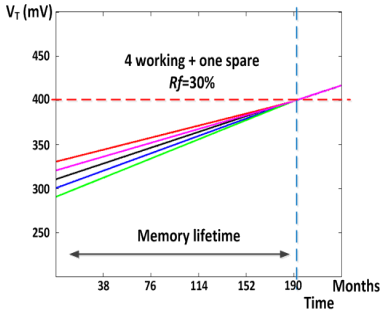

Figure 3. Adaptive proactive reconfiguration among 5 memory coulumns has resulted to uniform activity distribution and value convergance.

By using Matlab simulations, we compare the lifetime of our proactive proposal with a non-proactive reconfiguration scheme. We randomly generate fresh V_T values for a set of SRAM memory columns under normal distribution, with a given mean and standard deviation values (300mV and 30mV, respectively in our numerical examples). The maximum acceptable V_T aging value (H) before cell failure (when any dynamic parameter or the static noise margin, SNM, reach an unacceptable level) is assumed 400mV. The wear-out recovery factor is assumed at three different levels: 0%, 30% and 50%, in order to include

different technologies. Moreover, we have considered different number of active memory columns and performed 1000 Monte Carlo simulations. Table II compares both proactive technique's lifetimes in respect to a non-proactive scenario. As the results show, the adaptive technique can enhance the lifetime significantly.

TABLE II. MEMORY LIFETIME EXTENSION IN FUNCTION OF THE RF VALUES, THE APPLIED ALGORITHM AND STRUCTURES, IN COMPARISON WITH NON-PROACTIVE CONFIGURATION.

Configuration	IBM	Adaptive
4+1	1.4X @ Rf=0% 2.3X @ Rf=30% 3.5X @ Rf=50%	1.8X @ Rf=0% 3.2X @ Rf=30% 5X @ Rf=50%
8+1	1.3X @ Rf=0% 2.1X @ Rf=30% 3.1X @ Rf=50%	1.5X @ Rf=0% 2.7X @ Rf=30% 4.2X @ Rf=50%
16+1	1.1X @ Rf=0% 1.8X @ Rf=30% 2.4X @ Rf=50%	1.3X @ Rf=0% 2.2X @ Rf=30% 3.3X @ Rf=50%

VI. PROACTIVE ARCHITECTURE, MONITORING CIRCUITS AND TEST PROCEDURE

In this section, we depict our proposed adaptive proactive architecture inside a SRAM cache memory system. Then, we define our test procedure and circuits to monitor the variability and aging of SRAM cells in the columns and finally the control configuration and the implementation cost evaluation.

A. Architecture

We consider an 1kB SRAM memory example consisting of 128 columns divided into 8 groups of 16 columns and each memory column contains 64 6T-cells. We assume that the 1kB memory contains 8 spare columns and each one of the spare columns belongs to each group. All memory bit-lines (BL and BLB) are connected to a 1-bit bus that links them with the monitoring circuit (see Fig. 4). The word-lines coming from the row decoder are labeled as WL (WL_0-WL_{63}), and the test word-lines (those activated at testing phase) are named WL_T (WL_{T0}-WL_{T63}), and controlled by the reconfiguration controller. In this context, Fig. 4 presents our adaptive proactive memory architecture. Each set contains 17 columns (16 functional and 1 spare), with the added circuits and units required to perform the test and reconfiguration.

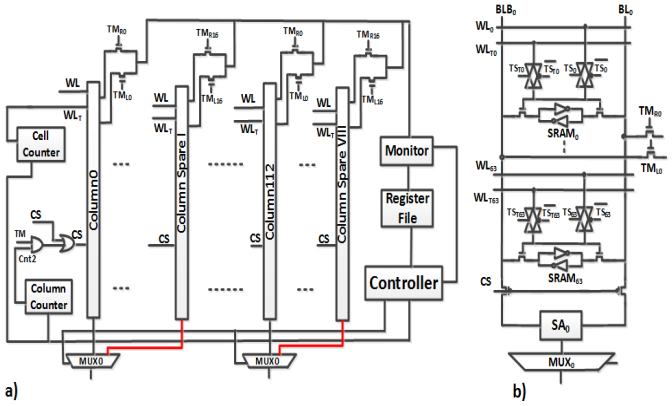

Figure 4. Architecture used for implementation of the adaptive proactive reconfiguration technique in an 1kB SRAM cache.

We use a single circuit to monitor all the columns of the memory, this avoids the impact of relative variability in the monitor circuit itself and results in a low overhead. Note that larger memory blocks can also be constructed by the architecture shown in Fig. 4.a, where a defined partition of columns can share the same monitoring circuit. Fig. 4.b depicts the structure of one memory column (column0) in our proposed approach. The column (BL and BLB) is connected to a monitoring circuit by the TM_{L0} and TM_{R0} transistors, and they are activated independently in the test phase. Two transmission gates and control signals (TS_x, TS_{Tx}) isolate the column memory cells from the undesired word-line during the normal memory operation and the monitoring phase. In this sense, the testing process does not interfere with the normal operation of the other memory columns (active columns). Next, we describe our monitoring technique.

B. Monitor and Test

In order to evaluate the time-zero variation and aging status of the SRAM cells in a cache memory array, we use an on-chip monitoring circuit. Note that, there exist some different approaches to measure the degradation of SRAM cells [14-15-16]. However in this work, we propose a novel and efficient monitoring approach that can measure the BTI (both NBTI and PBTI) wear-out and variability status of the individual SRAM memory cells in each memory column in a DC manner. Our proposed technique monitors the SRAM cells degradation in a column-by-column sequence. This strategy has no effect on the normal

memory operation, since it is applied when the specified column is in recovery mode, and it is disconnected during the normal operation of the column.

Our monitoring circuit is based on two current mirrors, which are connected to the memory column bit-lines. They track the current passing by each SRAM transistor and since the current value depends on the device status, in this way we can also analyze the process variation among the SRAM transistors and among all SRAM cells. Fig. 5 shows an example scheme of our proposed monitoring circuit for a SRAM cell. The test process for each column starts when the column goes to recovery mode and it requires two steps. The first one is to write a logic value '1' in all the SRAM storage nodes. Then, a counter enables each word-line (WL_T) one-by-one in order to measure the aging and process variability value of the pull-down transistors (the right NMOS, NR) and pull-up transistors (the left PMOS, PL) in SRAM cells of specific column. As an example, to monitor the NR transistor aging in the first column (column0) in Fig. 5, the controller enables the switches TM_{R0} and T_4 (the switch that selects the appropriate current mirror in respect to the N or PMOS), selects the input2 (the voltage value is generated by current mirror and the resistor) from Mux_1, and input2 (voltage generated by digital current source and the resistor) from Mux_2. Then it enables switches TM_{L0} and T_1 and selects input1 from Mux_1 and Mux_2 to monitor the aging in PL. Next, a '0' is written to all the SRAM storage nodes in the column, and again the counter turns on each word-line (WL_T) one-by-one and this time the monitoring circuit tracks the degradation in (NL) and (PR) transistors of SRAM cells. To avoid short channel effects and assure a good match of mirrored current, the devices used to measure the aging performance, TM and the current mirror transistors, are designed as long and wide channel devices. Note that since our monitoring circuit is applied to all the columns to order them based on their aging value, the possible mismatch or deviation does not have significance in the columns ordering.

We have designed and simulated our monitoring technique by using PTM (Predictive Technology Model) transistor [17] in HSPICE. As an example, we have assumed a logic value '1' is written in the SRAM storage node, and we monitor the aging in NR transistor of the SRAM cell. The dotted red line inserted in Fig. 5 illustrates the measurement path to test and monitor the performance of this specific SRAM transistor. The I_{NR} current is mirrored in the current mirror, and when the current I_{dig} gets equal to I_{CS2} the comparator output changes its state and the current value is recorded in the register file. The digital current source (I_{dig}) is shown in the inset of Fig. 5 and is implemented by a current mirror with elemental sources, and has a resolution of 8 bits. The measured current flows through the path made by SRAM pull down NR, access transistor (AC1) and the test switch (TM_{R0}) device.

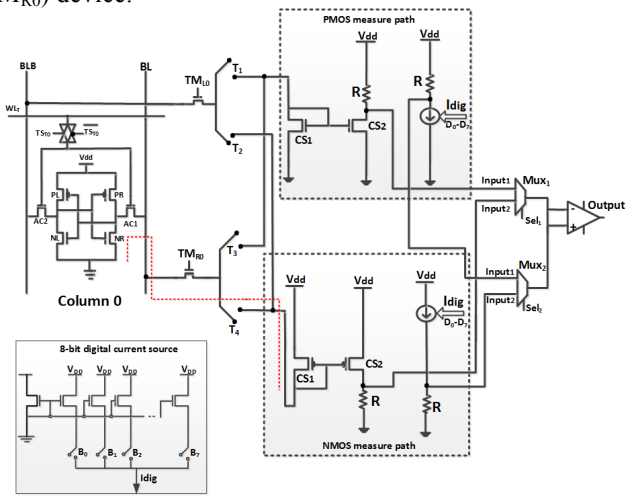

Figure 5. Process variation and aging sensing scheme in column 0,the inset shows the digital current source implemented by the current mirrors

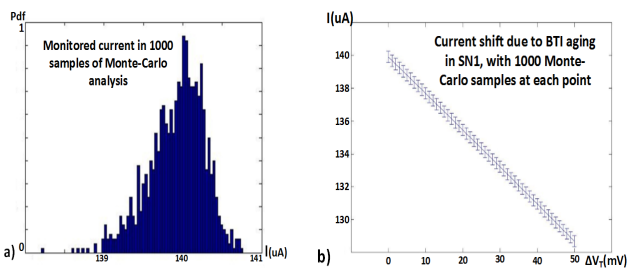

Figure 6. Monte-Carlo simulation considering variability in access transistor, b) Current decrease during V_T shift, representing aging of a NMOS

The TM_{R0} transistor is designed with larger size than the SRAM cell transistors, and only switches during the cell´s test mode, so it is slightly affected by aging and process variations. The access transistor (AC1) is also not affected by aging because it only switches when the cell is accessed from the decoder to read or write into the cell. Therefore, the important transistor to monitor aging in the path is the SRAM transistor (NR). Furthermore, when the current flows in this branch at the monitoring phase, the AC1 and TM_{R0} transistors are at their linear region while the (NR) transistor is in saturation region, so the measured current value will highly depend on the NR transistor current, and not on the access and TM_{R0} transistors. Fig. 6.a presents the current value in the branch, after a 1000 sample Monte-Carlo analysis and considering only variability in AC1 and fixed aging in NR transistors. The result shows that the current is slightly modified around the I_{SN1} (140uA), and it is the NR transistor that has the biggest impact on the current in the measurement path. Finally, Fig. 6.b shows the degradation in a NMOS device, and the relation between their V_T-shift and current weakening, after 1000 Monte-Carlo simulation at each aging point. It is observed that as the NMOS is stressed the V_T starts to shift down from its nominal value, and the device current reduces in accordance with it. Also it demonstrates that the current swing due to aging in NR is large enough and slightly affected by the process variation in access transistor, therefore we can order the columns correctly in respect to their aging.

C. Control and Evaluation Results

In order to manage this monitoring circuitry a control scheme is required, Fig. 7 shows the block diagram of the proactive reconfiguration control unit.

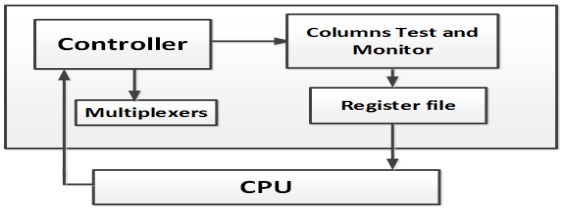

Figure 7. Reconfiguration control scheme.

The CPU itself is included in the control saving area, it reads the digital values of BTI in each memory column, and computes the appropriate adaptive recovery periods of each column. Each column and the corresponding spare one in that set is connected to a 2-1 multiplexer (Fig. 4.a). The controller is a built-in state machine, which controls the switching of the columns between active and recovery mode and outputs the correct column by controlling the multiplexers.

We evaluate our implementation in terms of silicon area overhead through area estimation. The digital units (register file, counters and the state machine) were coded in VHDL and synthesized with RTL compiler toward CMOS 45nm LP (low power) technology library [18]. Table III states the area overhead estimation, the area of our monitoring circuit implementation and the overall proactive reconfiguration monitoring design. Consider that the overall extra circuit implementation requires a silicon area of 670 um^2, which is around 12% of the 1kB 45nm SRAM silicon area [18].

TABLE III. ADAPTIVE PROACTIVE MONITORING DESIGN AREA IN SRAM

Component	Area μm^2
1kB 45nm 6T SRAM	6000 [18]
Controller, register file and counters	350
Our monitoring circuit	320 [19]
Overall monitoring components	670 (12%)

In comparison with the IBM approach our approach has only the additional overhead of the monitoring circuits. Our implemented methodology in the SRAM memory has slight effect on the memory cache performance. In each column reconfiguration step, the CPU copies the working column's data that goes into recovery mode in the spare column and this copied data is written back in the column before the next column reconfiguration step, with no impact on normal operation. One complete proactive reconfiguration of all the memory columns can take up to couple of the days and the frequency of reconfiguration process among the columns is very low, which allows the copying process to have enough time, therefore the small performance loss would be only at the switching time of a column to another, and the monitoring process of the recovery column can be a DC measurement.

VII. CONCLUSIONS

This work presents an adaptive proactive reconfiguration technique for SRAM-based memory systems. We analytically show that our adaptive proactive approach extends the system lifetime larger than the former (IBM) proactive approach. Moreover, we have proposed a specific monitoring circuit that tracks the time zero process variation and BTI aging of SRAM cells during

operation. With an adaptive proactive reconfiguration it is possible to extend the memory lifetime between 2X to 5X with around a 12% area overhead and negligible drop of performance.

REFERENCES

[1] S. Ghosh, K. Roy, "Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era," *IEEE Proc*, 2010, pp 1718-1751.

[2] S. Mukhopadyay, Q. Chen, K. Roy, "Memories in Scaled technologies: A Review of Process Induced Failures, Test methodologies, and Fault Tolerance," *Proc DECS*, 2007, pp. 1-6.

[3] A. Bansal, R. Rao, J-J. Kim, S. Zafar, J. Stathis "Impacts of NBTI and PBTI on SRAM static/dynamic noise margins," *Microelectronics Reliability*, 2009, pp. 642–649.

[4] K.N. Ganapathy, A.D. Singh, D.K Pradhan "Yield optimization in large RAM's with hierarchical redundancy," *IEEE Journal of Solid-State Circuit*, vol. 26, 1991, pp. 1259-1264.

[5] J. Shin, V. Zyuban, P. Bose, T.M. Pinkstone "A Proactive Wearout Recovery Approach for Exploiting Microarchitectural Redundancy to Extend Cache SRAM Lifetime," *Proc of 35th ISCA*, 2008, pp. 353-362.

[6] L. Li, Y. Zhang, J. Yang, "Proactive Recovery for BTI in High-*K* SRAM Cells," *Proc DATE*, 2011, pp. 1-6.

[7] R. Degraeve, M. Aoulaiche, B. Kaczer, Ph. Roussel, T. Kauerauf, S. Sahhaf, G. Groeseneken, "Review of reliability issues in high-k/metal gate stacks," *Proc IPFA*, 2008, pp. 1–6.

[8] S. Zafar, Y.H. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, M. Chudzik, "A comparative study of NBTI and PBTI (Charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, re gates," *Proc VLSI Tech*, 2006, pp. 23–25.

[9] T. Grasser, B. Kaczer, P. Hehenberger, W. Gos, R. O'Connor, H. Reisinger, W. Gustin, C. Schunder, "Simultaneous Extraction of Recoverable and Permanent Components Contributing to Bias-Temperature Instability," *Proc IEDM*, 2007, pp. 801-804.

[10] P.Pouyan, E. Amat, A. Rubio,"Process-Variability Aware Proactive Reconfiguration Technique for Mitigating Aging Effects in Nano-scale SRAM Lifetime," *Proc VTS*, 2012, pp 240-245.

[11] J. Hicks; et al. "45nm transistor reliability", Intel Technology Journal, vol. 12 (2), 2008, pp. 131-142.

[12] I. Kim, Y. Zorian, G. Komoriya, H. Pham, F.P. Higgins, J.L. Lewandowski, "Built in Self Repair for Embedded High Density SRAM,"*Proc Test*, 1998, pp. 1112-1119.

[13] www.mathworks.com

[14] K. Kang, M.A. Alam, K. Roy "Characterization of NBTI induced temporal performance degradation in nano-scale SRAM array using IDDQ," *Proc Int. Test*, 2007, pp. 1-10.

[15] Z. Qi, W. Jiajing, A. Cabe, S. Wooters, T. Blalock, B. Calhoun, M. Stan, "SRAM-Based NBTI/PBTI Sensor System Design," *Proc DAC*, 2010, pp 849-852.

[16] F. Ahmed, L Milor, "Reliable Cache Design with On-Chip Monitoring of NBTI Degradation in SRAM Cells using BIST," *Proc VTS*, 2010, pp 63-68.

[17] "Predictive Technology Models (PTM) [On line]," Http://ptm.asu.edu.

[18] S. Barasinski, L. Camus, S. Clerc, "A 45nm single power supply SRAM supporting low voltage operation down to 0.6V,"*Proc ESSCIRC*, 2008,pp.502-505.

[19] K. Lingkai, L. Yue, E. Alon, "A Multi-GHz Area-efficient Comparator with Dynamic Offset Cancellation," *Proc CICC*, 2011, pp.1-4.