# Process Variations and Process-Tolerant Design

Swarup Bhunia*, Saibal Mukhopadhyay$, and Kaushik Roy$
*Dept of EECS, Case Western Reserve University, $Dept of ECE, Purdue University,
skb21@case.edu; <sm, kaushik>@ecn.purdue.edu

***Abstract*—While CMOS technology has served semiconductor industry marvelously (by allowing nearly exponential increase in performance and device integration density), it faces some major roadblocks at sub-90nm process nodes due to the intrinsic physical limitations of the devices. One of the major barriers that the CMOS devices face at nanometer scale is increasing process parameter variations. Due to limitations of the fabrication process (e.g. sub-wavelength lithography and etching) and variations in the number of dopants in the channel of short channel devices, device parameters such as length (L), width (W), oxide thickness ($T_{ox}$), threshold voltage ($V_{TH}$) etc. suffer large variations. Variations in the device parameters, both systematic and random, translate into variations in circuit parameters like delay and leakage power, leading to loss in parametric yield. To deal with increasing parameter variations, it is important to accurately model the impact of device parameter variations at circuit level and develop process-tolerant design techniques for both logic and memory. This article analyzes the impact of process parameter variations on logic circuits and memory and focuses on some major works in the area of process-tolerant design methodology at circuit/architecture level.**

## I. INTRODUCTION

For the past four decades, aggressive scaling of conventional CMOS technology has enabled the semiconductor industry to meet its ever-increasing demand for computation power and integration density. However, at nanometer-scale geometry, VLSI system design faces some major challenges [1]. Manufacturing process parameter variation has emerged as a major bottleneck to efficient design of VLSI systems in sub-90nm CMOS technologies [2, 3]. Process imperfections due to sub-wavelength lithography and device level variations in small-geometry devices such as random dopant fluctuations and line edge roughness are making the devices exhibit large variations in their circuit parameters, particularly in the threshold voltage ($V_{th}$). Threshold voltage is a strong determinant of the circuit speed: low $V_{th}$ chips are typically faster than high-$V_{th}$ ones (since low $V_{th}$ corresponds to higher drive current). Statistical variations in device parameters lead to a statistical distribution of $V_{th}$. Consequently, delay of a circuit (and thus the maximum allowable frequency of operation) also follows a statistical distribution. Moreover, threshold voltage variation poses concern in robustness operation, particularly in Static Random Access Memory (SRAM) and dynamic logic circuits (such as domino).

Process variation is usually classified into two categories: die-to-die or inter-die, which is variation across different dies; and within-die or intra-die, which is variation among transistors within each die [2]. Die-to-die (D2D) variation changes the performance corner (fast or slow) of a particular die. The variation affects each transistor in the die in a systematic way; that is, if a die is in the high $V_T$ (slow) corner, all transistors will have high $V_T$. On the other hand, within-die (WID) variation affects each transistor differently resulting in transistors with different $V_{th}$'s within close proximity due to effects such as random dopant fluctuations, line-edge roughness, or channel length variations.

Increasing variations (both inter-die and intra-die) in device and interconnect parameters (channel length, gate width, oxide thickness, device threshold voltage etc.) produce large spread in the speed and power consumption of integrated circuits (ICs) [3, 4]. Consequently, parametric yield of a circuit (probability to meet the desired performance or power specification) is expected to suffer considerably, unless an overly pessimistic worst-case design approach is followed. Since leakage power of a circuit has exponential dependence on device threshold voltage ($V_{th}$), parameter variations results in large variability in leakage power [15, 22] along with variation in circuit delay. Fig. 1(a) shows the die-to-die variations in $V_{th}$, which is normally distributed with a 3σ value of 30mV at 180nm CMOS process technology. Fig. 1(b) plots the distribution of operating frequency and leakage current over a large number of high-end micro-processor chips [3]. From the figure, it can be observed that a high-performance design may suffer from about 30% variation in maximum allowable frequency of operation (Fig. 1b) and about 20X variation in leakage power due to variations in transistor parameters [3]. Wide spread in the leakage power distribution has emerged as another important cause of yield loss due to bound on static power dissipation [22]. For dynamic logic circuits, such as domino, threshold voltage variations cause an additional impact on variation in noise margin, which consequently affects robustness of operation.

Since worst-case design approach may incur prohibitive design overhead, multitude of research efforts have been devoted to explore alternative design methodologies under variations. Broadly, three classes of techniques are proposed to ensure/enhance yield under variations while incurring minimal impact on design overhead: ***1) Statistical design approach***, where a circuit parameter (e.g. delay or leakage) is modeled as a statistical distribution (e.g. Gaussian) [4, 7, 9, 13-14] and the circuit is designed to meet a constraint on yield (or to maximize it) with respect to a target value of the parameter [4-6, 8, 11, 15, 19]. Gate sizing or dual-$V_{th}$ assignment are typically used to vary circuit delay or leakage distribution. ***2) Post-Silicon***

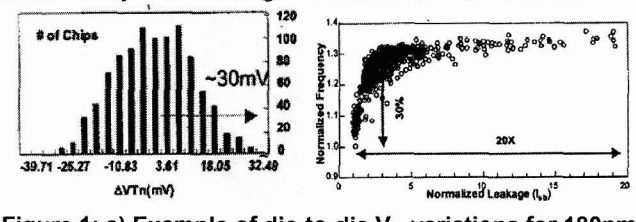



**Figure 1: a) Example of die-to-die $V_{th}$ variations for 180nm CMOS process [3]; b) Leakage and frequency variations of a high performance design [3].**

20th International Conference on VLSI Design (VLSID'07)
0-7695-2762-0/07 $20.00 © 2007 IEEE

***compensation and correction***, where parameter shift is detected and compensated/corrected after manufacturing by changing operating parameters such as supply voltage, frequency or body bias. Post-silicon techniques, such as adaptive body-biasing [18] or frequency scaling can affect both power and performance of a circuit. 3) ***Variation avoidance***, where a given circuit is synthesized in such a way that the delay failures due to variations can be identified in run time and avoided by adaptively switching to two-cycle operations [23].

In case of memory, the inter-die and intra-die variations in process parameters (in particular, threshold voltage) can lead to large number of failures in a Static Random Access Memory (SRAM) array, thereby, degrading the memory design yield in nanometer technologies [16-17]. To improve parametric yield of nano-scaled memories, different circuit and architectural level techniques can be used [12, 17, 24]. In this article, we will briefly analyze different SRAM failures due to parameter variations and next, we will describe two different self-repairing techniques– at the circuit level, using adaptive body biasing and at the architecture level, using Built-in-self-test (BIST), redundancy and address remapping. The discussed self-repair mechanisms can improve yield much beyond what can be achieved using conventional techniques such as row/column redundancy and Error Correcting Codes (ECC) alone.

The rest of the article is organized as follows. Section II presents some major techniques for process-tolerant logic design. In this part, we briefly analyze the impact of process parameter variations on logic circuits and then focus on three broad categories of process-tolerant logic design. In Section III, we analyze process-induced failures in SRAM and then propose robust memory design solutions with respect to process parameter variations. Section IV concludes the article.

## II. PROCESS VARIATION IN LOGIC

As mentioned before, process parameter variations play increasingly important role in circuit marginalities with technology scaling and hence, pose a major design concern. In logic circuits, parameter variation causes parametric yield loss since a circuit designed at nominal process corner may fail to satisfy delay and/or leakage target under parameter variations [3-4, 6]. Conventional wisdom dictates a conservative design approach (e.g., scaling up the $V_{DD}$ or upsizing logic gates) to avoid a large number of chip failures due to variations. However, such techniques come at the cost of considerable increase in power and/or die area. Over the past few years, therefore, researchers have looked for alternative design paradigms to ensure yield under process variations with minimal impact on design overhead. Statistical design approach has been widely investigated as an effective method to ensure yield under process variations. On the other hand, design techniques for post-silicon correction or compensation of process-induced failures have been proposed as efficient alternative solutions.

Due to quadratic dependence of dynamic power of a circuit on its operating voltage, supply voltage scaling has been extremely effective in reducing the power dissipation. Researchers have investigated logic design approaches that are robust with respect to process variations and, at the same time, suitable for aggressive voltage scaling. Design optimization techniques using gate sizing and dual-$V_{th}$ assignment to improve power typically increase the number of critical paths in a circuit, giving rise to the so-called "wall effect" [10], which in turn causes concern in terms of

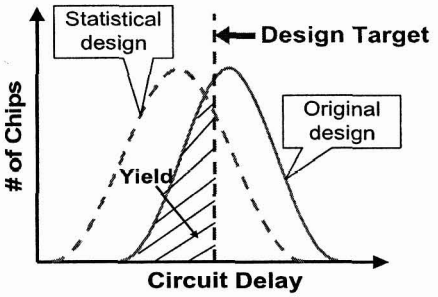


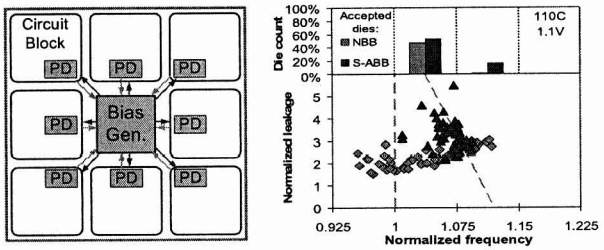

**Figure 2: Delay distribution of a circuit before and after statistical design. The yield, computed as the probability of meeting a delay target, is considered as an objective or constraint of the design optimization process.**

parametric yield. The uncertainty-aware design technique [10] describes an optimization process to reduce the wall effect although it does not address the problem of power dissipation. Next, we will discuss some major techniques on process-tolerant design that fall into three broad categories.

### A. Statistical design and modeling

With inter-die and intra-die parameter variations, traditional approach to static analysis and design of circuit considering specific target frequencies and power budgets (dynamic + leakage power) in mind becomes less effective. Considering a $V_{th}$ distribution as shown in Fig. 1a, circuit delay and power also can be modeled as a statistical distribution. In recent years, statistical analysis of timing and power have been extensively explored [4, 7, 9, 13-14]. Several parametric yield models have been proposed to consider impact of different sources of variations on circuit delay and power [4, 11]. An important challenge associated with determining statistical delay and leakage distribution for a circuit has been computing signal (such as delay, arrival time) correlation accurately and propagating them across logic level. On the other hand, statistical design methodology that either ensures or enhances certain parametric yield (e.g. with respect to delay) under specific design constraint (e.g. on area or power) has been addressed by many researchers [4-6, 8, 11, 15, 19]. Gate-level sizing and/or $V_{th}$ assignment have been primarily used as a tool to modulate the circuit delay distribution for yield improvement or yield-constrained area/power minimization. In a statistical design, circuit delay is typically modeled as a Gaussian distribution (Fig. 2); timing yield is modeled as the probability to meet the target delay (shaded region in Fig. 2); and the delay distribution is changed in a way to improve yield during the design optimization process.

Under process parameter variations, delay of a pipelined circuit follows a statistical distribution. Operating frequency of a pipelined circuit is determined by the delay of the slowest pipeline stage. However, under statistical delay variation the slowest stage is not readily identifiable and estimation of the pipeline yield with respect to a target delay is a challenging problem. An interesting observation is made in [4] that changes in the logic depth and imbalance between the stage delays can improve the pipeline yield. A novel statistical methodology is described to optimally design a pipeline circuit for enhancing yield under an area constraint. Once the independently optimized stages are combined to form a pipeline, a final global optimization step is proposed to improve pipeline yield with no area overhead, based on a concept of *area borrowing*. Optimization results show




(a) (b)

**Figure 3: a) Adaptive body biasing scheme considering within-die delay variations; b) leakage vs. frequency distribution of an adaptive body biasing scheme that considers both inter- and intra-die variations [18].**

that, incorporating proper imbalance among the stage areas in a 4-stage pipeline improves design yield up to 16% for equal area over a balanced design [4].

Profitability of a design is conventionally equated with yield. However, large spread in the frequency distribution due to increasing uncertainties has led to the concept of speed-binning to improve the design profit [19]. Presently, speed-binning is widely used during manufacturing test to qualitatively sort the working (i.e. free from manufacturing defects) ICs based on their highest allowable frequency of operation. Since high-frequency ICs correspond to higher price points compared to their low-frequency counter parts, maintaining yield at a target circuit delay (i.e. frequency) under statistical delay distribution does not ensure high profit.

Considering a price profile that determines the price point for an IC corresponding to its performance, it can be easily shown that two different delay distributions (with different mean and standard deviation) of ICs can result in the same yield but significantly different profit. A statistical gate sizing approach for profit optimization is proposed in [19] that shows average profit enhancement by 19% for a set of benchmark circuits.

## B. Post-Silicon Process Compensation/Correction

In this category of solutions, process variation is detected using on-chip process sensor or manufacturing test and deviation of circuit parameters due to variation is compensated/corrected by appropriate technique. One such technique, called RAZOR uses dynamic detection and correction of circuit timing errors to tune processor supply voltage [25]. It is applicable to systems that allow dynamic voltage/frequency scaling for power reduction.

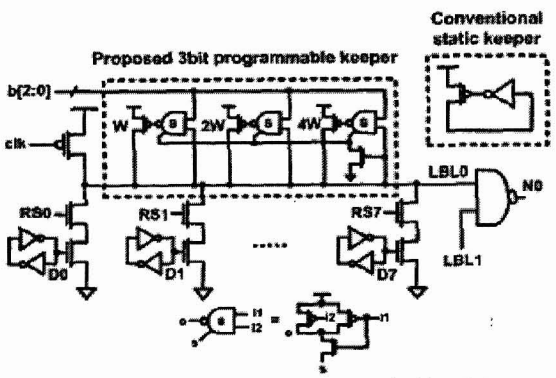



**Figure 4: Register file with process compensating dynamic circuit technique (digitally programmable keeper size can be configured to be: 0, W, 2W, ... , 7W) [20].**

Razor relies on a combination of architectural and circuit-level techniques for efficient error detection and correction of delay path failures by using a shadow latch, controlled by a delayed clock, corresponding to each flipflop in the design. In a given clock cycle, if the combinational logic meets the setup time for the main flip-flop, then both main flip-flop and shadow latch will latch the correct data. If combinational logic does not complete computation in time, the main flip-flop will latch an incorrect value, while the shadow latch will latch the late-arriving correct value. The error flag is raised prompting restoration of the correct value from the shadow latch.

Body bias can control leakage and performance of a die and thus has been investigated as a process adjustment tool. While forward body bias (FBB) helps to improve performance in active mode (by lowering the $V_{th}$), reverse body bias (RBB) is effective to reduce leakage power (by increasing the $V_{th}$). A practical application of body bias to adjust process shift requires accurate detection of process shift at different parts of a circuit and application of optimal body bias voltage which maximizes the performance under power constraint. A bidirectional adaptive body bias (ABB) technique is used to compensate for die-to-die parameter variations in [18] by applying an optimum pMOS and nMOS body bias voltage to each die. To account for intra-die variations, an enhancement of this technique is proposed that requires a phase detector (PD), that determines frequency of a block from its critical path replica, in each circuit block and the central bias generator considers output of all PDs to determine the optimal bias (Fig. 3a). Measurement results show that the technique results in an increase in number of acceptable dies (Fig. 3b) as well as number of high-frequency dies.

A process variation compensating technique for dynamic circuits is described for sub-90nm technologies where leakage variation is severe [20]. Increasing $I_{OFF}$ with process scaling has forced designers to upsize the keeper in dynamic circuits to obtain an acceptable robustness under worst-case leakage conditions. However, large (over 20x) variation in die-to-die NMOS $I_{OFF}$

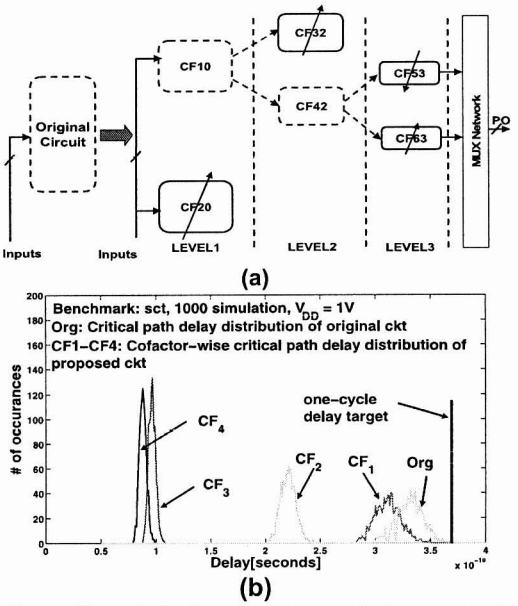



**Figure 5: a) Hierarchical expansion and sizing of cofactors resulting in b) desired path delay distribution for failure avoidance (shown for a benchmark circuit) [23].**

indicates that 1) a large number of low leakage dies suffer from the performance loss due to an unnecessarily strong keeper, while 2) the excess leakage dies still cannot meet the robustness requirements with a keeper sized for the fast corner leakage. In [20], the authors describe a Process-Compensating Dynamic (PCD) circuit technique that improves robustness and delay variation spread by restoring robustness of worst-case leakage dies and improving performance of low-leakage dies. Unlike prior fixed-strength keeper techniques, the keeper strength is optimally programmed based on the respective die leakage. Fig. 4 shows the PCD scheme with a digitally programmable 3-bit keeper applied on an 8-way register file local bitline (LBL). Such a keeper enables 10% faster performance, 35% reduction in delay variation, and 5x reduction in robustness failing dies over conventional static keeper design in 90nm dual-$V_{th}$ CMOS process [20].

### C. Avoidance of Variation-Induced Failures

Realizing low power design in scaled technologies is often a challenging task due to increased parametric failures under process parameter variations. Robustness with respect to variations and low power operations typically impose contradictory design requirements. Low power design techniques such as voltage scaling, dual-$V_{th}$ etc. can have a large impact on parametric yield. A novel paradigm to design low-power circuits under process variation has been proposed in [23]. The proposed paradigm, based on the concept of *critical path isolation*, makes a circuit amenable to aggressive voltage scaling, while being robust to parametric failures. This is accomplished by a synthesis technique that 1) isolates and predicts the set of possible paths that may become critical under process variations, 2) ensures they are activated rarely, and 3) tolerates any delay failures in the set of critical paths by adaptively operating in two-cycles (assuming all standard operations are single cycle). This allows us to operate the synthesized circuit at reduced supply voltage while achieving the required yield. Fig. 5a shows the partitioning and gate sizing steps and Fig. 5b plots path delay distribution for a MCNC benchmark after the proposed synthesis process. The delay margin between critical and non-critical blocks helps to avoid delay failure and achieve voltage scaling.

The notion of *critical path isolation* indicates confinement of critical paths of a synthesized design to a known logic block or cofactor. This is accomplished by partitioning a circuit into multiple cofactors using Shannon decomposition and then using gate-sizing to create timing margin between cofactors. Any delay errors (that may occur under a single cycle operation) are predicted dynamically by decoding a small set of inputs and are adaptively avoided with two cycle operations. Simulation results on benchmark circuits show promising results in power saving (about 60% on average) compared to conventionally synthesized and sized netlist with an average area overhead of 18% [23].

## III. PROCESS VARIATIONS IN MEMORY

Die-to-die and within-die variations in process parameters result in the mismatch in the strengths of different transistors in an SRAM cell (Fig. 6), which causes functional failures. The functional failures due to parametric variations degrade the memory yield (i.e. the number of non-faulty chips) [17]. The principal reason for parametric failures is the intra-die variation in threshold voltage of the cell transistors due to random dopant fluctuations [16-17]. However, intra-die parameter variation also has a strong impact on the failure probability of a cell. In

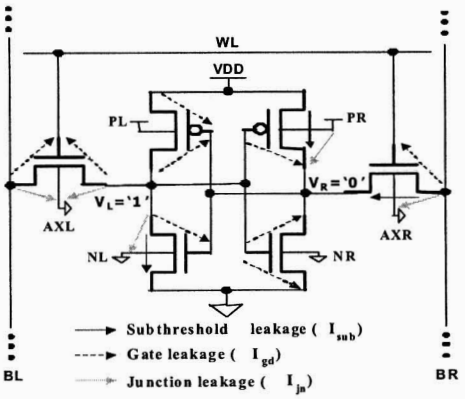



**Figure 6: SRAM Cell storing "0" at node R.**

particular, low-$V_{th}$ dies suffer mostly from read and hold failures while high-$V_{th}$ dies suffer from access and write failures [17].

Conventionally, redundant rows/columns [26] are used in memories to improve yield. These redundancy techniques have limitation on the number of faulty rows/columns it can handle, due to resource limitation and design complexity. In particular, the failures due to random dopant effect are randomly distributed across the dies, resulting in a large number of faulty rows/columns. Recovery from such defects is difficult to handle by row/column redundancy alone. Moreover, SRAM failures due to process variation change depending on operating condition (e.g. supply voltage, frequency). The operation condition changes dynamically, which poses challenges to any static techniques, such as row/column redundancy. ECC [21], employed in existing design to correct transient faults (such as soft error), can also be used to correct failures due to process variations. However, ECC has limitations on the number of error bits it can correct and it incurs considerable area and performance overhead.

With the limitation of the existing fault tolerant methods, SRAM, that can repair itself and, hence, reduce the number of failures would be very effective for yield improvement in nano-scale technologies. In this article, we will discuss two major self-repairing techniques at the circuit and architecture level: 1) a circuit technique that senses the process corner of the die and applies adaptive body bias to improve yield; 2) an architecture technique that uses Built-In-Self-Test (on-line or off-line) to determine a failure map of the memory, re-maps the address based on the failure map, and improves yield at the cost of little performance/power overhead.

### A. SRAM Parametric Failures

The parametric failures in an SRAM cell due to process variations are principally due to [21]:

***Read Failure*** - Flipping of the SRAM cell data while reading. The read failure can be reduced by increasing the difference between the voltage rise at the node storing "0" while reading (say, $V_{READ}$) and the trip-point of the inverter ($V_{TRIPRD}$) associated with the node storing "1".

***Write Failure*** – Unsuccessful write to the SRAM cell. Write failure occurs if the node storing "1" cannot be discharged through the access transistors during the word-line turn on time.

***Access Failure*** –Access failure occurs if the voltage differential between the two-bitlines is below the offset voltage of the sense-amplifier at the time of the sense-amplifier firing. It occurs due to the reduction of the bit-line discharging current


0-7695-2762-0/07 $20.00 © 2007 IEEE

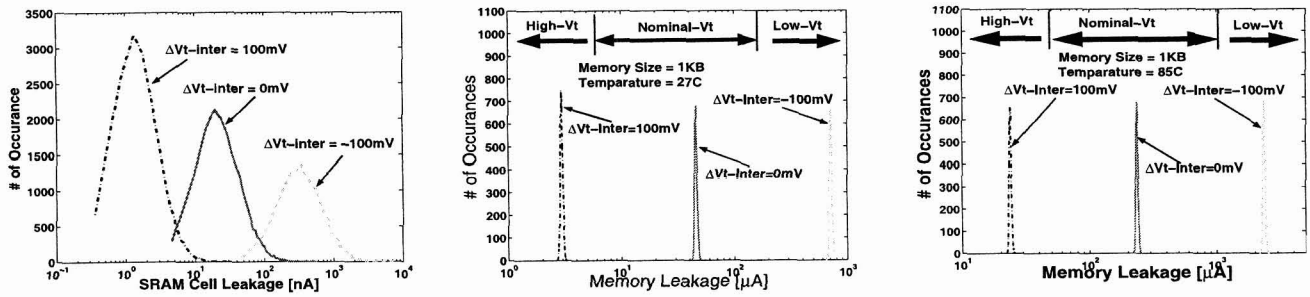



**Figure 7: Effect of intra-die $V_{th}$ at different inter-die $V_{th}$ corners of SRAM: a) leakage distribution (due to intra-die variation) of an SRAM cell; b) leakage distribution (due to intra-die variation) of the 1KB SRAM array at T=27°C; and (c) at T=85°C [21].**

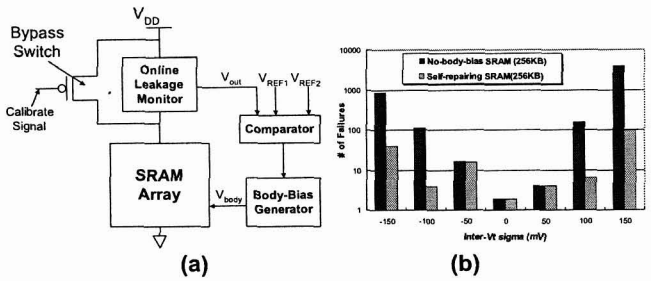



**Figure 8: a) Self-repairing SRAM scheme; b) reduction in number of failures in 256KB memory array [24].**

through the access and pull-down NMOS transistors.

***Hold Failure*** - The destruction of the cell data in the standby mode with the application of a lower supply voltage. The hold failure occurs due to high-leakage of the NMOS transistors connected to the node storing "1". At a lower VDD, due to the leakage of the NMOS, the node storing "1" reduces from VDD (which is enhanced by a weak PMOS). If that voltage becomes lower than the trip-point of the inverter storing "0" the cell flips.

## B. Self-Repair Using ABB

A $V_{th}$ shift toward low $V_{th}$ process corners, due to inter-die variation, increases the read and the hold failures of SRAMs (Fig. 7a). This is because of the fact that, lowering the $V_{th}$ of the cell transistors increases $V_{READ}$ and $V_{TRIPRD}$, thereby increasing read failures [21]. The negative $V_{th}$ shift increases the leakage through the transistor $N_L$, thereby, increasing the hold failures. On the other hand, for SRAM arrays in the high-$V_{th}$ process corners, the probabilities of access failures and write failures are high (Fig. 7a). This is principally due to the reduction in the current drive of the access transistors. The hold failure also increases at the high $V_{th}$ corners, as the trip-point of the inverter PR-NR increases with positive $V_{th}$ shift. Hence, the overall cell failure increases both at low and high-$V_{th}$ corners and is minimum for arrays in the nominal corner (Fig. 7a). Consequently, the probability of memory failure is high at both low-$V_{th}$ and high-$V_{th}$ process corners (Fig. 7b).

Let us now discuss the effect of the body-bias (applied only to NMOS) on different types of failures. Application of reverse body-bias increases the $V_{th}$ of the transistors which reduces $V_{READ}$ and increases $V_{TRIPRD}$, resulting in a reduction in the read failure (Fig. 7c) [21]. The $V_{th}$ increase due to RBB also reduces the leakage through the NMOS thereby reducing hold failures (Fig. 7c). However, increase in the $V_{th}$ of the access transistors due to RBB increases the access and the write failures. On the other hand, application of FBB reduces the $V_{th}$ of the access transistor, which reduces both access and write failures. However, it increases the read ($V_{READ}$ increase and $V_{TRIPRD}$ reduces) and hold (leakage through NMOS increases) failures (Fig. 7c) [21].

To determine the correct body bias to apply to the SRAM chip for failure probability improvement, the process corner, in which the memory chip sits, needs to be determined. An effective way to perform $V_{th}$ binning is to use leakage monitoring. The random intra-die variation in threshold voltage results in significant variation in cell leakage, particularly, the subthreshold leakage. In a self-repairing SRAM using "Leakage Monitoring", the measured leakage is compared with the reference currents to identify the inter-die process corner of the chip. Based on this measurement, the right body bias is applied to the chip. The schematic of a self-repairing SRAM array with self-adjustable body-bias generator is shown in Fig. 8a. Experimental results on self-repairing circuit on reduction in number of failures (as shown in Fig. 8b) appear promising.

## C. Architecture Level Technique

An architecture level technique proposed in [12] detects and replaces faulty cells by adaptively remapping the cache. This architecture assumes that the cache is equipped with a built-in-self-test (BIST) unit, which tests the entire cache and

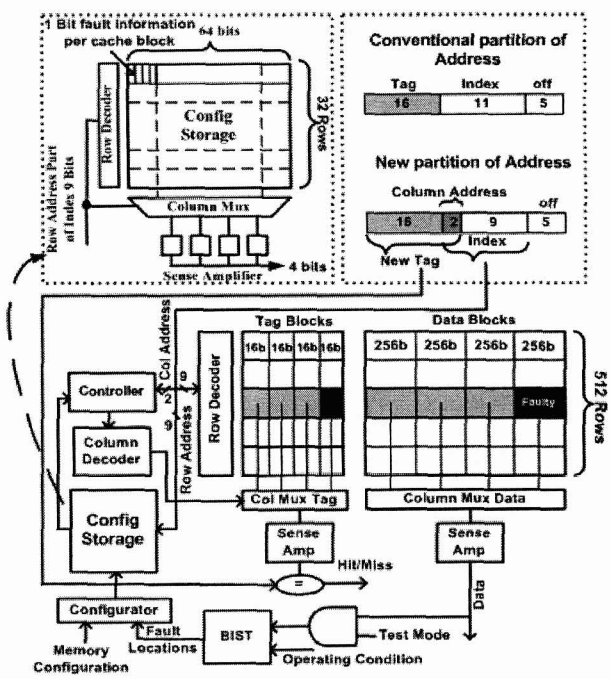



**Figure 9: Architecture of 64K process-tolerant cache [12].**


0-7695-2762-0/07 $20.00 © 2007 IEEE

detects faulty cells due to parameter variations. Fig. 9 shows the anatomy of the process-tolerant cache architecture. The scheme downsizes the cache by forcing the column MUX to select a non-faulty block in the same row if the accessed block is faulty. This scheme maps the whole memory address space into the resized cache in such a way that the remapping is transparent to the processor. Hence, the processor accesses the resized cache with the same address as in a conventional architecture.

Conventionally, a cache is divided into cache blocks (e.g. 256 bit per block, Fig. 9) and several cache blocks are stored in a single row. A block is considered faulty even if a single cell in a block is faulty. BIST detects the faulty blocks and feeds the information into the configuration storage. Configuration storage is a small memory, which stores 1-bit fault information per cache block. In conventional cache access, all the blocks in a cache row are selected simultaneously by a single wordline. Finally, column MUX chooses one block based on column address. The key idea of the proposed architecture is to force the column MUX to select another block in the same row, if the accessed block is faulty. In the proposed architecture, both cache and *config* storage is accessed in parallel. The *config* storage provides the fault information to the controller about all the blocks in the accessed row. Based on this fault information, controller forces the column MUX to select another non-faulty block (if the accessed block is faulty) in the same row by altering the column address, thereby resizing the cache. In case of a faulty block access, this scheme selects the first available non-faulty block. Hence, as long as there is one non-faulty block in each row, this architecture can correct any number of faults.

Simulation results using BPTM 45nm technology show that in best case this architecture can handle up to 75% faulty blocks with only 1.8% and 0.5% energy and area overhead, respectively. The cache access time remains unchanged, although there is a performance hit due to increased miss rate at diminished cache size. However, the yield improves by up to 94% compared to a yield improvement of 34% using redundant rows/columns.

## IV. Conclusions

In this paper, we discuss the failure mechanisms in logic circuits and SRAM due to inter- and intra-die process parameter variations and review some major circuit/architecture level design solutions to address process variation related yield loss. The impact of process parameter variations on circuit functionality has been well-understood and modeling of circuit parameter such as delay, leakage, SRAM stability etc. under process variations has been well-established. Conventional deterministic design methods have been shown ineffective or inefficient under variations and a paradigm shift from deterministic to probabilistic design has been proposed. On the other hand, existing fault tolerant methods does not work well to correct process-induced failures in both logic circuit and memory. Thus, novel techniques for process correction/compensation followed by process detection have been widely investigated.

This article refers to only a limited number of published works from the vast set of existing research on analysis of process variations and design for process tolerance at circuit/architecture level. The techniques for process-tolerant design should come with negligible overhead in terms of performance, power and area and at the same time be effective to improve parametric yield significantly. We believe that with process variations becoming more and more pronounced in the scaled technology generations, on-going research on process-tolerant design will continue to produce novel low-overhead and robust design solutions for upcoming nanometer technologies.

## References


[1] International Technology Roadmap for Semiconductors, http://public.itrs.net, 2004.

[2] K. A. Bowman et al., "Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration", *IEE JSSC*, 2002, pp. 183-190.

[3] S. Borkar et al., "Parameter Variations and Impact on Circuits and Micro-architecture", *DAC*, 2003, pp. 338-342.

[4] A. Datta et al., "Delay modeling and statistical design of pipelined circuit under process variations", *IEEE TCAD*, 2006, pp. 2427-2436.

[5] E. T. A. F. Jacobs and M. R. C. M. Berkelaar, "Gate Sizing Using a Statistical Delay Model", *DATE*, 2000, pp. 283-290.

[6] S. Choi et al., "Novel Sizing Algorithm for Yield Improvement under Process Variation in Nanometer Technology", *DAC*, 2004, pp. 454-459.

[7] K. Kang et al., "Statistical Timing Analysis using Levelized Covariance Propagation", *DATE*, 2005, pp. 764-769.

[8] A. Agarwal et al., "Circuit Optimization using Statistical Timing Analysis", *DAC*, 2005, pp. 321-324.

[9] H. Chang and S. S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations using a Single PERT-like Traversal", *ICCAD*, 2003, pp. 621-625.

[10] X. Bai et al., "Uncertainty-Aware Circuit Optimization", *DAC*, 2002, pp. 58-63.

[11] M. Mani et al., "An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints", *DAC*, 2005, pp. 309-314.

[12] A. Agarwal et al., "A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies", *IEEE TVLSI*, 2005, pp. 27-38.

[13] P. Fox et al., "Statistical analysis of propagation delay in digital integrated circuits", *IEEE ISSCC: Digest of Technical Papers*, 1972, Vol. XV, Feb 1972, pp. 66-67.

[14] M. C.-T. Chao et al., "Static Statistical Timing Analysis for Latch-Based Pipelined Designs", *ICCAD 2004*, pp. 468-472.

[15] A. Srivastava and D. Sylvester, "A General Framework for Probabilistic Low-Power Design Space Exploration considering Process variation", *ICCAD 2004*, pp. 808-813.

[16] A. Bhavnagarwala et al., "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," IEEE JSSC, pp. 658-665, April 2001.

[17] S. Mukhopadhyay et al, "Statistical design and optimization of SRAM for yield enhancement," ICCAD, 2004, pp. 10-13, Nov. 2004.

[18] J.W. Tschanz et al, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," IEEE JSSC., vol. 37, no. 11, Nov, 2002, pp. 1396-1402.

[19] A. Datta et al, "Speed Binning Aware Design Methodology to Improve Profit under Parameter Variations," *ASP-DAC*, 2006, pp. 1545-1554.

[20] C. H. Kim et al., "A Process Variation Compensating Technique for Sub-90nm Dynamic Circuits", *Symp. on VLSI Circuits*, June 2003, pp. 205 – 206.

[21] S. Mukhopadhyay et al, "Modeling and estimation of failure probability due to parameter variations in nano-scale SRAMs for yield enhancement," *Symp. on VLSI Circuits*, 2004, pp. 64 – 67, June 2004.

[22] R. Rao, et al, "Parametric yield estimation considering leakage variability," *DAC*, pp. 442 - 447 June, 2004.

[23] S. Ghosh et al, "A New Paradigm for Low-power, Variation-Tolerant and Adaptive Circuit Synthesis Using Critical Path Isolation," *to appear ICCAD*, 2006.

[24] S. Mukhopadhyay et al, "Reliable and self-repairing sram in nano-scale technologies using leakage and delay monitoring," *ITC*, Nov 2005, pp. 1126-1135.

[25] D. Arnst et al., "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. MICRO-36*, 2003, pp. 7–18.

[26] K. Itoh, VLSI Memory Chip, Springer, 2001.