Design in the Nanometer Regime: Process Variation

Kaushik Roy
S. Mukhopadhyay, C. Kim, H. Mahmoodi, S. Ghosh, A. Datta, A. Agarwal

Electrical & Computer Engineering
Purdue University
Exponential Increase in Leakage

Silicon Micro-electronics

\[ \frac{I_{ON}}{I_{OFF}} = 10^6 \]

Subthreshold Leakage

Gate Leakage

Source

Drain

n+ n+

Gate

Junction Leakage

Bulk

Silicon Nano-electronics

\[ \frac{I_{ON}}{I_{OFF}} = 10^3 \]

Non-Silicon Technology

\[ \frac{I_{ON}}{I_{OFF}} \sim 10^{2-6} \]

Leakage Power (% of Total)

Must stop at 50%

A. Grove, IEDM 2002

Leakage Power (% of Total)

Technology (μ)

1.5 0.7 0.35 0.18 0.09 0.05
Variation in Process Parameters

Device parameters are no longer deterministic

Inter and Intra-die Variations

Random dopant fluctuation

Device parameters are no longer deterministic
Reliability

Temporal degradation of performance -- NBTI

- Defects
- Life time degradation
- Failure probability
- Tech. generation
- Time
Substantial variation in leakage across dies
4X variation between nominal and worst-case leakage
Performance determined at nominal leakage
Robustness determined at worst-case leakage
On-Chip Memories: Process-Tolerance/ Self-Repair (Post-Si)
Memories: On-chip SRAMs

- On-chip memory size is increasing with scaling
- Challenges: **Leakage and Variability**

![Graph showing the percentage of chip area taken by SRAM cache](image-url)
Global and Local Variations

\[ \delta V_t = \Delta V_{t-\text{GLOBAL}} + \delta V_{t-\text{LOCAL}} \]

Random Dopant Fluctuation
Parametric Failures in SRAM

Parametric failures can degrade SRAM yield

- Read Failures
- Write Failures
- Access Failures
- Hold Failures

Test & Repair using Redundancy

Faulty chips

Working chips
Process Variations in On-chip SRAM

\[ \sigma_{Vt} \approx 30\text{mv}, \text{ using BPTM 45nm technology} \]

Simulation of an 64KB Cache

A. Agarwal, et. al, JSSC, 05

Parametric failures \( \rightarrow \) Yield degradation
Inter-die Variation & Cell Failures

![Diagrams showing inter-die variation and cell failures]

- **Low-Vt Corners**
  - Read failure $\uparrow$
  - Hold failure $\uparrow$

- **High-Vt Corners**
  - Access failure $\uparrow$
  - Write failure $\uparrow$

**Inter-die Vt shift** ($\Delta V_{th\text{-GLOBAL}}$)

**GLOBAL** $\sigma$
Post-Silicon Repair: Proposed Approach

Apply correction to the global variation to reduce number of failures due to local variations
Memory failure probabilities are high when inter-die shift in process is high.
Self-Repairing SRAM Array

Reduce the dominant failures at different inter-die corners to increase width of low failure region.
Self-Repairing SRAM Array

Reduce the dominant failures at different inter-die corners to increase width of low failure region

Region A
Region B
Region C

RBB
ZBB
FBB

Inter-Die Vt Shift [V]
Array Leakage Monitoring

\[ Y = \sum_{i=1}^{N} X_i \Rightarrow \frac{\sigma_Y}{\mu_Y} = \frac{1}{\sqrt{N}} \frac{\sigma_X}{\mu_X} \]

- Adding a large number of random variables reduces the effect of intra-die variation

Leakage of entire SRAM array is a reliable indicator of the inter-die Vt corner
Self-Repair using Leakage Monitoring

Entire array leakage is monitored to detect inter-die corner and proper body-bias is selected.
Test-Chip of Self-Repairing SRAM

Technology: IBM 0.13 μm
128KB SRAM
Dual-Vt Triple-well tech.
Number of Trans: ~ 7 million
Die size: 16mm²
VLSI CKT Symp. 2006, ITC 2005

Simulation results for 1MB array designed in IBM 0.13μm
Quantized (3 Level: FBB, ZBB, RBB) body bias scheme is a cost effective solution with good yield enhancement possibility.
Hold Failure Measurement

- Low Vt array shows more number of hold failures
- Application of reverse body bias to NMOS (RBB) reduces number of hold failures
Process Tolerance: Register Files (Post-Si)
Keeper upsizing degrades average performance
Process Compensating Dynamic Circuit Technology

3-bit programmable keeper

C. Kim et al. , VLSI Circuits Symp. ‘03

Opportunistic speedup via keeper downsizing
Leakage Binning Results

Output codes from leakage sensor
Robustness Squeeze

- **5X reduction in robustness failing dies**

![Bar chart showing normalized DC robustness with conventional and this work methods compared.](chart.png)
10% opportunistic speedup
Self-Contained Process Compensation

**Fab**

**Wafer test**

*Process detection*
- Leakage measurement
- On-die leakage sensor

*Correction*
- Program PCD, Body-bias, Source-bias, ..

**Customer**

**Package test**

**Burn in**

**Assembly**
On-Die Leakage Sensor For Measuring Process Variation

- High leakage sensing gain
- Compact analog design sharing bias generators

C. Kim et al., VLSI Circuits Symp. ’04
Self-Repair: Architecture Level (Design Time)
- BIST detects the faulty blocks
- Config Storage stores the fault information

Idea is to resize the cache to avoid faulty blocks during regular operation
Resizing the Cache

Config Storage is accessed in parallel with cache

Feeds the fault information to controller

Controller alters the column address

- Conventionally multiple blocks are stored in a row
- Column MUX selects one block based on column address

Force the column MUX to select a non-faulty block in the same row if the accessed block is faulty

Handle large number of faults without significantly reducing the cache size
Mapping Issue

More than one INDEX are mapped to same block

Tag does not match, cache miss

Include column address bits into TAG bits

Resizing is transparent to processor → same memory address
Effective Yield of 64K Cache

- ECC + Redundancy yield \( \sim 77\% \)
- Proposed architecture + Redundancy yield \( \sim 94\% \)
Proposed architecture can handle more number of faulty cells than ECC, as high as 890 faulty cells.

Saves more number of chips than ECC for a given $N_{\text{Faulty-Cells}}$. 
CPU Performance Loss

- Increase in miss rate due to downsizing of cache
- Average CPU performance loss over all SPEC 2000 benchmarks for a cache with 890 faulty cells is ~ 2%
Process Variation and Logic
Parameter Variations: Logic

Conventional ways to improve yield
  – improve power, reduce robustness
  – improve robustness, increase power
Logic: A New Paradigm for Low-Voltage, Variation Tolerant Circuit Synthesis Using Critical Path Isolation
Design Considerations for Low Power and Robust Circuit

- Few predictable critical paths
- Low activation probability of critical paths
- Slack between critical and non-critical paths under variations
Motivations

Features:
- Single critical path (activated by $P_0P_1P_2P_3=1$ & $C_{i,0}=1$)
- Low activation probability of critical path

<table>
<thead>
<tr>
<th>VDD</th>
<th>TCLK</th>
<th>Crit. path delay</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1V, 260ps</td>
<td></td>
<td>260ps</td>
<td>13uW (1-cycle)</td>
</tr>
<tr>
<td>0.8V, 260ps</td>
<td></td>
<td>330ps</td>
<td>7.4uW (rare 2-cycles, decoder)</td>
</tr>
</tbody>
</table>

44% power saving by reducing supply voltage and, operating critical path at 2-cycle and other paths at 1-cycle (adaptive stretching of clock)
Shannon’s Expansion Based Partitioning

\[ f(x_1, \ldots, x_i, \ldots, x_n) = x_i \cdot f(x_1, \ldots, x_i = 1, \ldots, x_n) + x'_i \cdot f(x_1, \ldots, x_i = 0, \ldots, x_n) \]

\[ = x_i \cdot CF_1 + x'_i \cdot CF_2 \]

\[ CF_1 = f(x_1, \ldots, x_i = 1, \ldots, x_n); \quad CF_2 = f(x_1, \ldots, x_i = 0, \ldots, x_n) \]

Activation probability of cofactors can be reduced
How to choose Control Variable?
Isolation and Slack Creation by Proper Expansion & Sizing

- Some non-critical paths are made faster to create a timing slack to facilitate supply scaling
Design at supply voltage of 0.85V leads to
- Average power saving ~60%
- Area overhead ~-8%
- Performance degradation ~6%
Test Chip and GDS Layout

Technology: 0.13 μm
Circuit size: 0.30 mm²
CMOS Design: Sizing

Combinational and sequential elements

- Statistical estimation of circuit delay and yield under process variation
- Delay of the slowest stage limits the highest clock rate

Under statistical delay variation how can we identify the slowest stage?
Balanced Stage Yields

Balanced design (uncorrelated stage delays)

• Equal stage delay and yield ($Y_0$), for 3 stage pipeline $Y_T = Y_0^3$

• Individual stage areas ($A_1, A_2, A_3$) are usually different

How to improve yield of a balanced design?

• Increase yield of each stage equally

→ Large area overhead
Unbalancing Stage Yields - I

Mean design delay

\[ Y_{\text{BALANCED}} = Y_0 \times Y_0 \times Y_0 \]

\[ Y_{\text{UNBALANCED}} = Y_1 \times Y_2 \times Y_0 \]

\( Y_1 \times Y_2 > Y_0^2 \Rightarrow \text{Improvement} \)

Unbalancing may improve yield
Careful unbalancing among the pipeline stages improves the design yield
Rule for Unbalancing Stages

Slope of area vs. delay curve at delay $T_D$:

$$R_i = \left| \frac{\partial A}{\partial D} \right|_i$$

For any stage $i$

if ($R_i > 1$)

reduce large stage area
small increase in delay

else

increase small stage area
large improvement in delay

- **Slope-based heuristic**

Applicable to heterogeneous pipelines with non-regular stages
Temporal Degradation: NBTI
Temporal Reliability Issues in CMOS Technology

- \textit{HCI} – Hot Carrier Injection
- \textit{NBTI} – Negative Bias Temperature Instability
  - Increase in $V_T$ of PMOS with time
  - The dominant reliability factors in scaled tech.
- \textit{TDDB}, etc.

- \textit{Need for sensors to detect and possibly correct NBTI induced failures in memory and logic}
- \textit{Sizing – slight over design to improve yield}
NBTI-aware design method

- **Over-design** is required to guarantee a lifetime stability of the circuit
- **LR sizing** is used to optimize the circuit
  - Size the circuit considering the worst-case $V_T$ degradation over the lifetime
Conclusion

• Process parameter variations and variations of device parameters over time are becoming very important in Nano-scale technologies
• Tools for estimation of yield, failure probability and performance degradation over time are required
• There is need to consider speed, yield and power dissipation for design of high speed memory systems
• Both design and post-Si techniques are required to improve yield of memories