SRAM supply voltage scaling: a reliability perspective

Joint work with my advisors at UCB:
Dr. Kannan Ramchandran
and Dr. Jan Rabaey

Animesh Kumar
Electrical Engineering
IIT Bombay, Mumbai

Outline

◊ Introduction and contributions

◊ SRAM cell’s failure modeling (90nm CMOS)

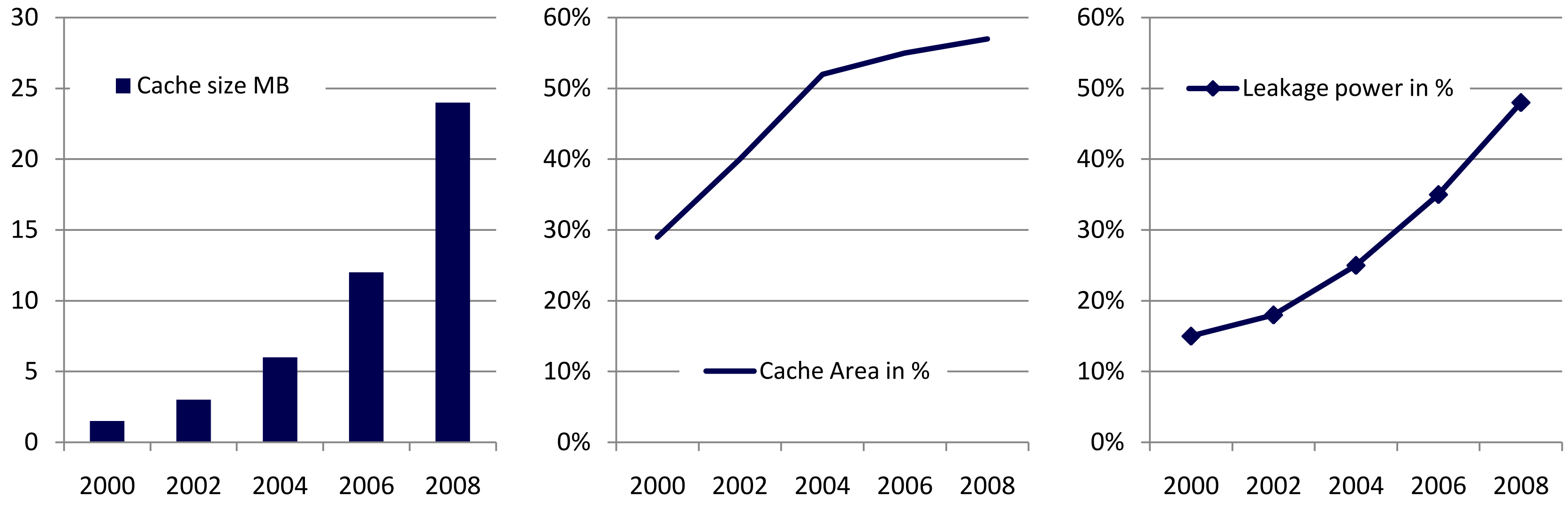
◊ Leakage-power optimization results (90nm CMOS)

◊ Conclusions

Outline

- ◊ Introduction and contributions
- ◊ SRAM cell’s failure modeling (90nm CMOS)
- ◊ Leakage-power optimization results (90nm CMOS)
- ◊ Conclusions

Cache/SRAM trends in microprocessors



Courtesy : Intel, Borkar et al.[2000]

For Intel microprocessors:

- ◊ Cache size, cache contribution to the total area, and leakage-power percentage in chip increase with technology or year
- ◊ As a result, cache leakage-power is a significant fraction of the total chip power

SRAM leakage in sensor nodes

For “mostly-idle” devices, SRAM leakage dominates the total power

- ○ Consider the Charm chip of picoRadio group designed for sensor nodes
- ○ SRAM is the largest block with 65% transistors
- ○ During standby, SRAM needs power supply to retain the data, while other blocks are turned off

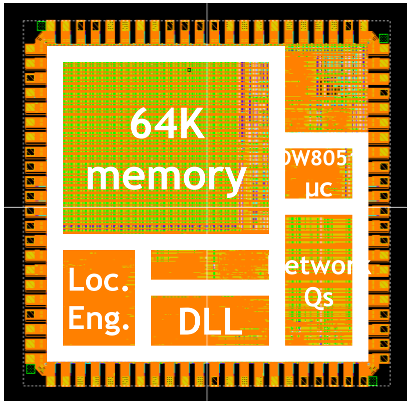


Figure: The Charm Chip [Sheets-Rabaey et al.]

◊ An obvious method to reduce the leakage-power is to reduce the supply voltage at which the data is stored in standby [Kim-Blauuw et al.’02, Qin-Rabaey et al.’04]

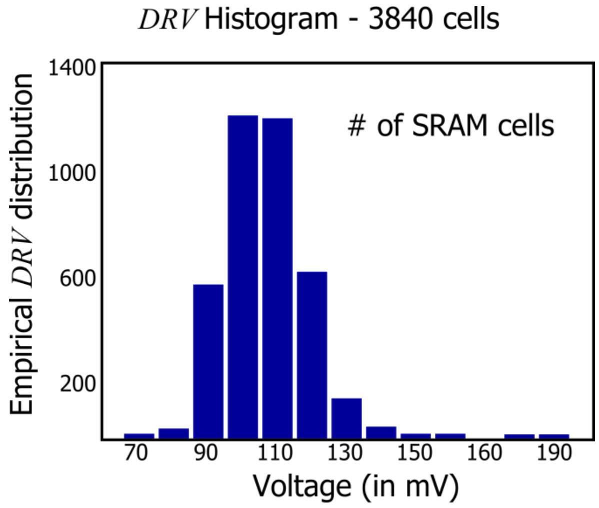
◊ How much voltage reduction is possible?

What stops cache supply voltage reduction?

Example: (Hold failures)

◊ An SRAM cell has a minimum supply voltage for preserving data and it is called the data-retention voltage (DRV) [Qin-Rabaey, et al.’04]

◊ Due to process-variations, DRV exhibits a range of values



[Qin-Rabaey et al. 2006]

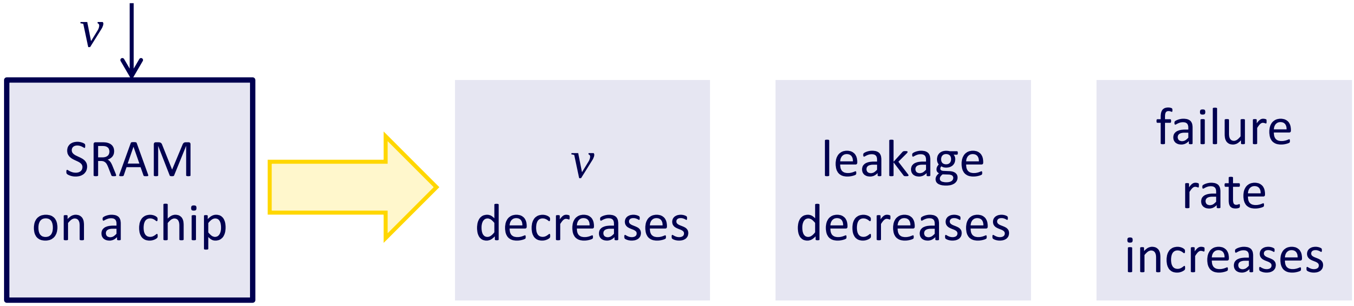
Main idea

◊ Standby mode: SRAM has to store data, but read or write activity is zero

◊ Standby voltage can be reduced, till storage failures happen

◊ The min. voltage needed to store all the bits determines the leakage power

SRAM cell failure mechanisms



Failure mechanisms

◊ Soft-errors [Hazucha-Svennsson’98, Degalahal et al.’05]

◊ Parametric failures – failures that affect read/write/hold [Roy et al.’06, Nassif-Agarwal’07]

◊ Supply noise induced errors [Nassif-Kozhaya’00, Alon et al.’05] → Tackled by a $100\mathrm{mV}$ margin

◊ Erratic fluctuations [Agostinelli et al.’05] → Ignored, lack of complete models

Parametric failures

◊ Destructive Read –> read-upset

◊ Unable to Write –> write-failure

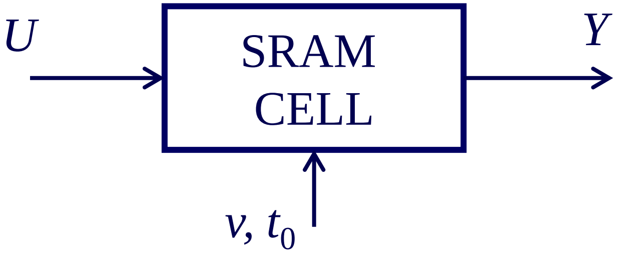
◊ Unable to Store –> hold-failure

◊ Write or read time is insufficient –> access-time/write-time failure

SRAM cell’s channel (probabilistic) model

SRAM cell model:

For a bit U that has been written into the SRAM cell, the following input-output probabilistic model will be used,



$Y =$ × (erasure), with prob. $p_x(v)$

$Y = U^c$ (error), with prob. $p_e(v)$

$Y = U$, otherwise

Cause of data-failure	*Type of fault*	*Cause*	*Probability*
Read/Write/Access/Hold	Erasure/spatial	Process variations	$p_x(v)$
Soft-error	Error/Temporal	Radioactivity	$p_e(v) = t_0\ r_s(v)$

Main ideas used in this work

Trade-off consists of the following steps:

◊ Reduce supply voltage aggressively

◊ Allow individual SRAM cells to be (statistically) more prone to failures

◊ Use system level techniques – error correction coding and data-refresh (scrubbing) – to compensate for the increase in per cell "failure"

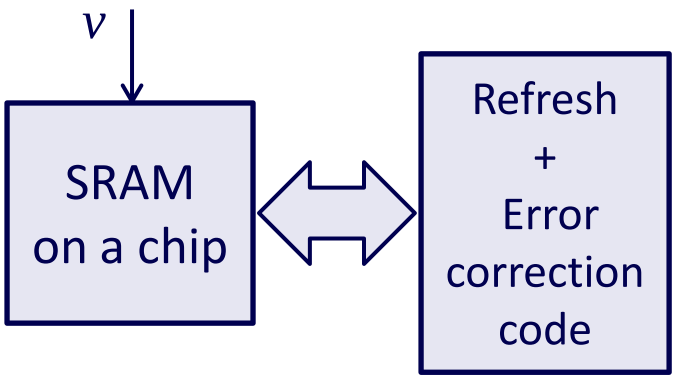
◊ Minimize leakage power, including coding overheads, over choice of supply voltage

Related work: Error correction codes in SRAM

◊ Mostly a parity check or a single-bit error correction code is used in cache for extra reliability [Spica-Mak'04, Slayman'05]

◊ [Heegard-El Gamal'83] studied capacity of memories; [Slayman'05] summarizes error-correction codes aspects of SRAM, etc.

Related work: data-refresh (or scrubbing)



Trade-offs while scrubbing:

◊ Mitigates bit-error accumulation due to soft-errors (or any other noise mechanism)

◊ Consumes extra power, which has to be accounted for, while reducing leakage

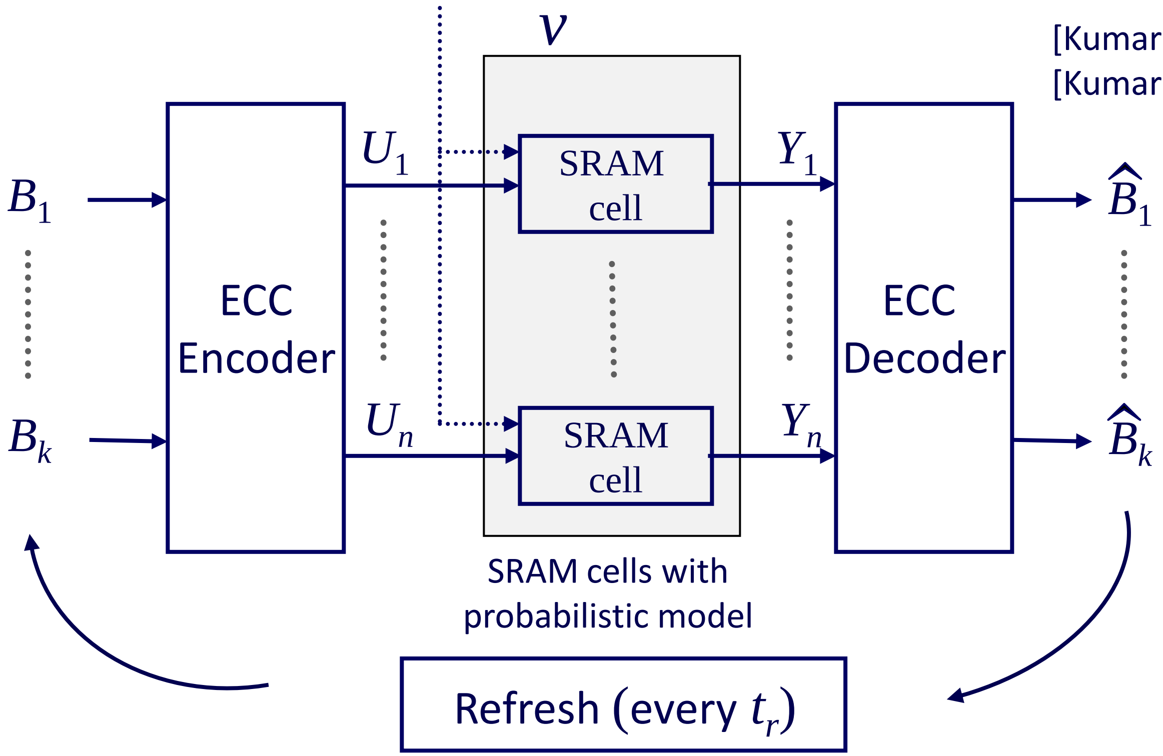
Research literature:

◊ Proposed for increasing reliability of memory/storage systems [Saleh et al.'90]

◊ Error probability improvement due to scrubbing have been calculated for selected error correction codes in the presence of soft-errors by [Bajura et al.'07]

Proposed low-leakage SRAM block diagram

[Kumar-Rabaey-Ramchandran'09]
[Kumar(Thesis)'08]



◊ $(B_1, \ldots, B_k)$ is the data-bit vector to be stored

◊ Input is encoded by a rate k/n, $n > k$ code to $(U_1, \ldots, U_n)$

◊ After each refresh time-period, $(Y_1, \ldots, Y_n)$ is decoded to $(\widehat{B}_1, \ldots, \widehat{B}_k)$ and restored

Contributions and performance metrics

- ◊ Trade-offs between leakage-power, error-correction code, error probability and data-refresh rate are presented using circuit level simulations (90nm CMOS) – principal “knob” being the supply-voltage
- ◊ Efficient statistical estimation method to obtain the circuit-level probabilistic model of SRAM cell is developed

Cost and constraint functions

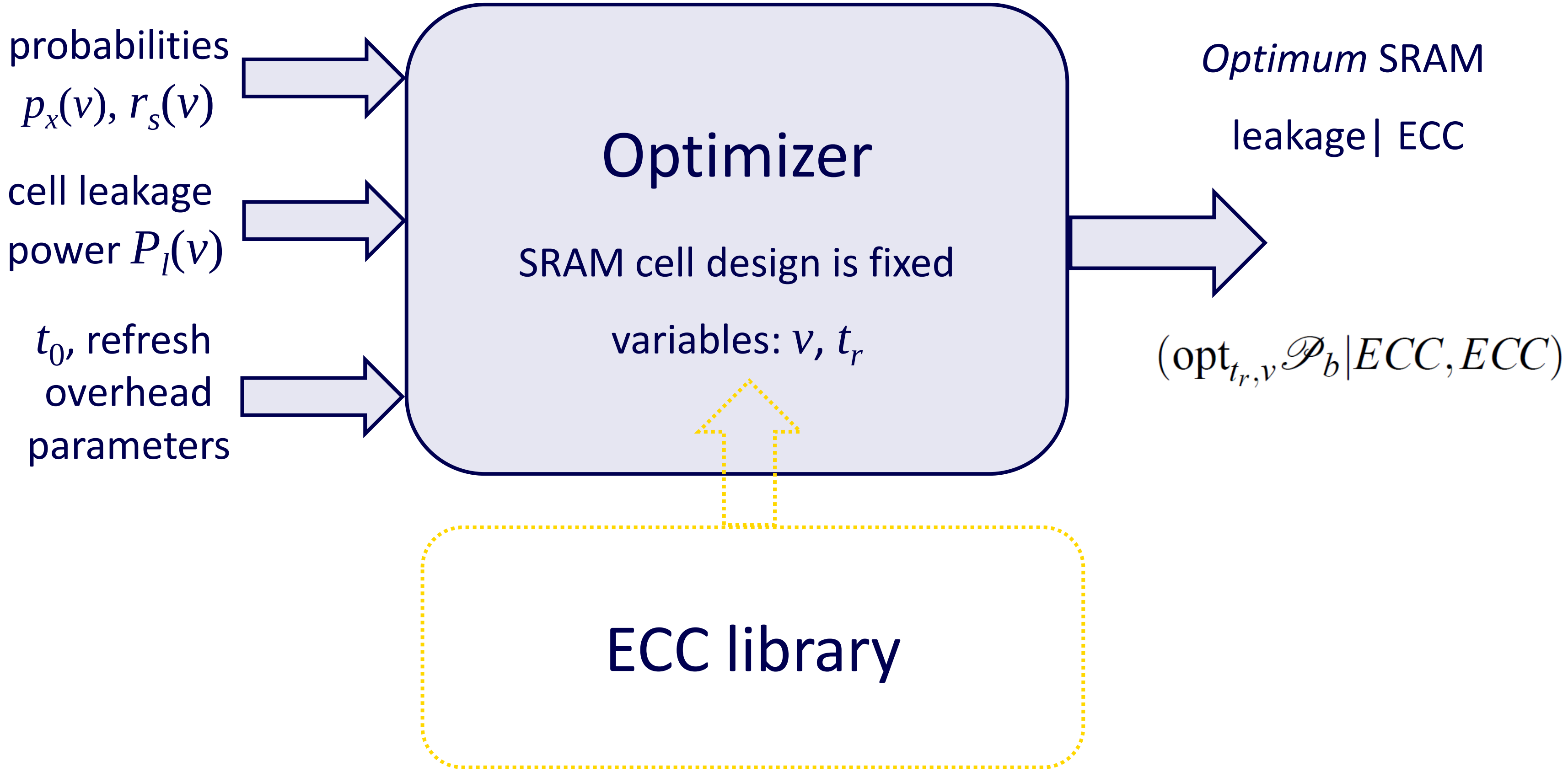
Cost: $$\mathscr{P}_b := \mathscr{P}_b(v, t_r, ECC) = \frac{n}{k} P_l(v) + \frac{\text{refresh energy}}{k t_r}$$

where $P_l(v)$ is the leakage power at voltage v

Constraint: Given ECC, choose t_r such that error probability at v is equal to error probability at $v = 1.0\text{V}$ with $[31, 26, 3]$ Hamming code

Optimization framework

Within the cost-constraint setup from the previous slide, the following separable optimization framework is developed [Kumar-Rabaey-Ramchandran'09], [Kumar(Thesis)'08]


probabilities $p_x(v), r_s(v)$
cell leakage power $P_l(v)$
t_0, refresh overhead parameters
Optimizer
SRAM cell design is fixed
variables: v, t_r
Optimum SRAM leakage| ECC
$(\text{opt}_{t_r,v}\mathscr{P}_b|ECC, ECC)$
ECC library

Circuit techniques to reduce leakage power

◊ Circuit level technique have also been proposed to reduce the leakage power

- Modification of SRAM cell's transistor parameters, e.g., [Zhao et al.'04, Qin-Rabaey et al.'08]
- Addition of sleep transistors or control gate, e.g., [Zhang et al.'05, Agarwal-Roy'03]
- Designing an asymmetric SRAM cell, e.g., [Azizi et al.'03]
- Usage of new transistors like FINFET, e.g., [Guo-King-Nikolic et al.'05]
- Using a different structure than 6-Transistor SRAM cells, e.g., [Calhoun-Chandrakasan'06, Ali-Faisal-Bayoumi'05]

◊ These techniques are "stackable" with the technique proposed in this talk

Outline

◊ Introduction and contributions

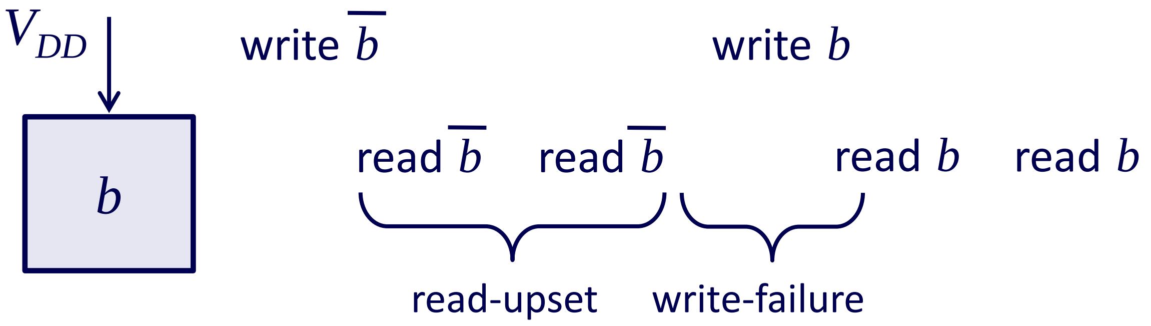
◊ SRAM cell’s failure modeling (90nm CMOS)

◊ Leakage power optimization results (90nm CMOS)

◊ Conclusions

Simplifying assumptions

- ◊ The failures are spatially independent
  - Achieved by interleaving at a negligible energy cost (distributes MBU) [Slayman'05, Blum et al.'07]
  - The parametric failures are spatially fixed, and hence they are determined as erasures or don't care (x) by test-patterns
  - Soft-errors are spatially/temporally random
- ◊ Soft-errors don't happen during read/write (negligible probability)



V_{DD}

b

write $\overline{b}$ write b

read $\overline{b}$ read $\overline{b}$ read b read b

read-upset write-failure

This test pattern reveals the location of parametric failures

SRAM channel model's constituents

- At v = 0.2V, the SRAM cell was not writeable. The supply voltage range to consider is 0.3V to 1.0V
- The parametric failure probability is the erasure probability

$$p_x(v) = p_{pf}(v) \leq p_r(v) + p_w(v) + p_h(v) + p_{at}(v) + p_{wt}(v)$$

- Soft-errors , and other noise-like failures have rates, which determine the error probability

$$p_e(v) \leq t_r\, r_s(v), \qquad p_e(v) << 1$$

Modeling of failures

- Why modeling?
  - Brute force Monte Carlo simulations will require more than trillion trials
  - Due to these complexity reasons, supply voltage is chosen in $100\mathrm{mV}$ steps
- **Soft-errors:**
  - Estimation with macro-models developed by Freeman’96
- **Parametric failure probabilities:**
  - Read upset – estimation with read-noise margin (RNM)
  - Write failure – estimation with write-noise margin (WNM)
  - Access-time failure – estimation with extreme value theory
  - Write-time failure – estimation with extreme value theory
  - Hold failure – negligible compared to read upset

Roy et al.’06

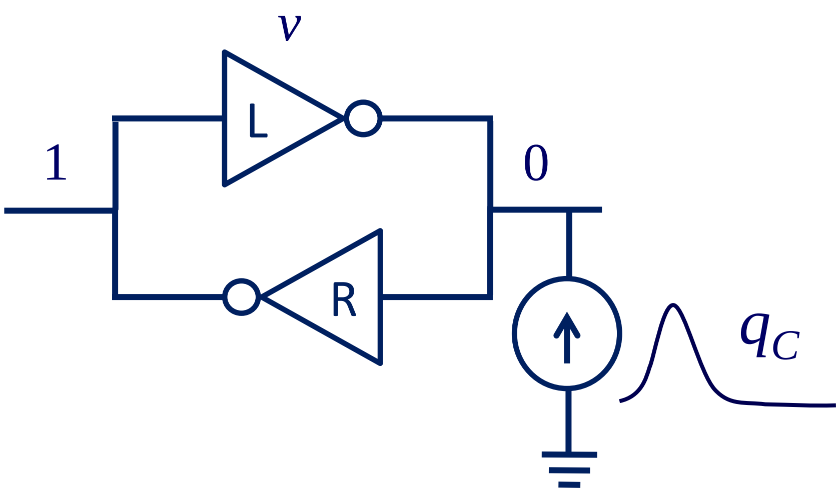
Nassif-Agarwal’07

Meindl et al.’01

Bhavnagarwala’05

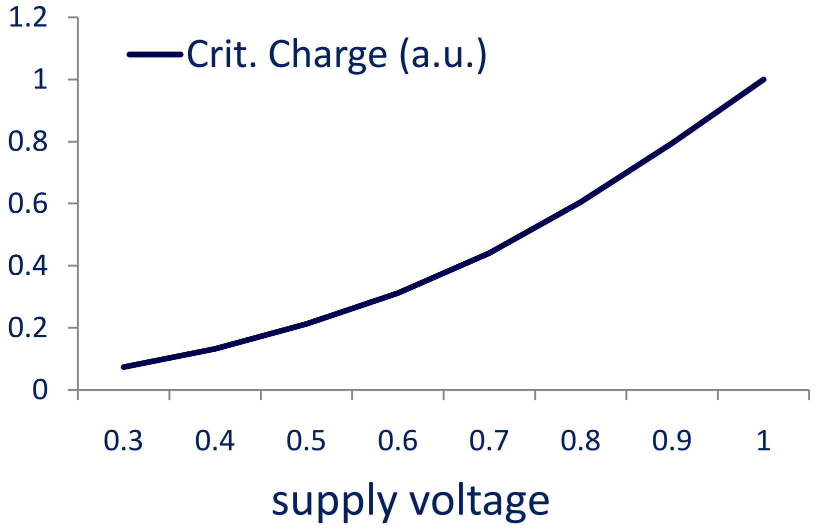
Kumar-Rabaey-Ramchandran’09

Soft-error rate estimation



Critical charge: Minimum charge q_C needed in the radiation induced current-pulse for flipping the SRAM-state

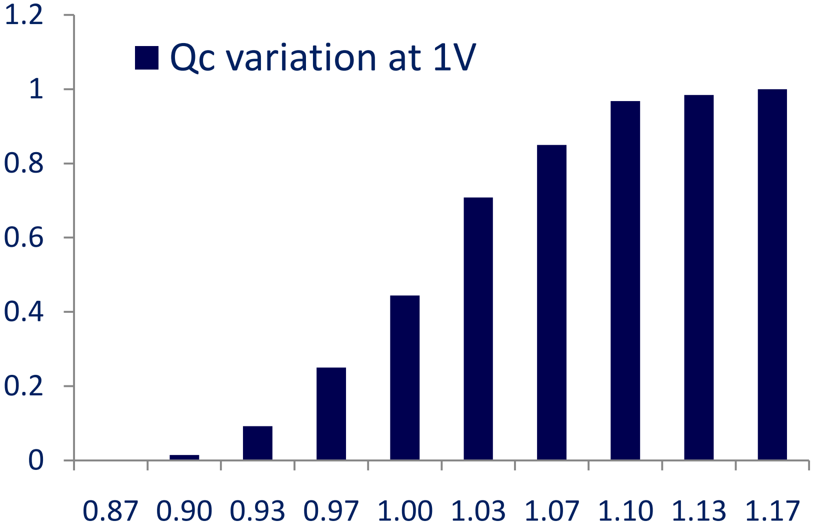
- [Freeman'96] [Hazucha'00]



$$\boxed{r_s(v) = K\exp(-\alpha q_C(v))}$$

α and K are obtained from the literature

Critical charge variation



Q_c (a.u.) cumulative distribution obtained for the 90nm CMOS tech

$$\boxed{E[R_s(v)] = E[K\exp(-\alpha Q_C(v))]}$$

The relative difference

$$(E[R_s] - r_s)/r_s$$

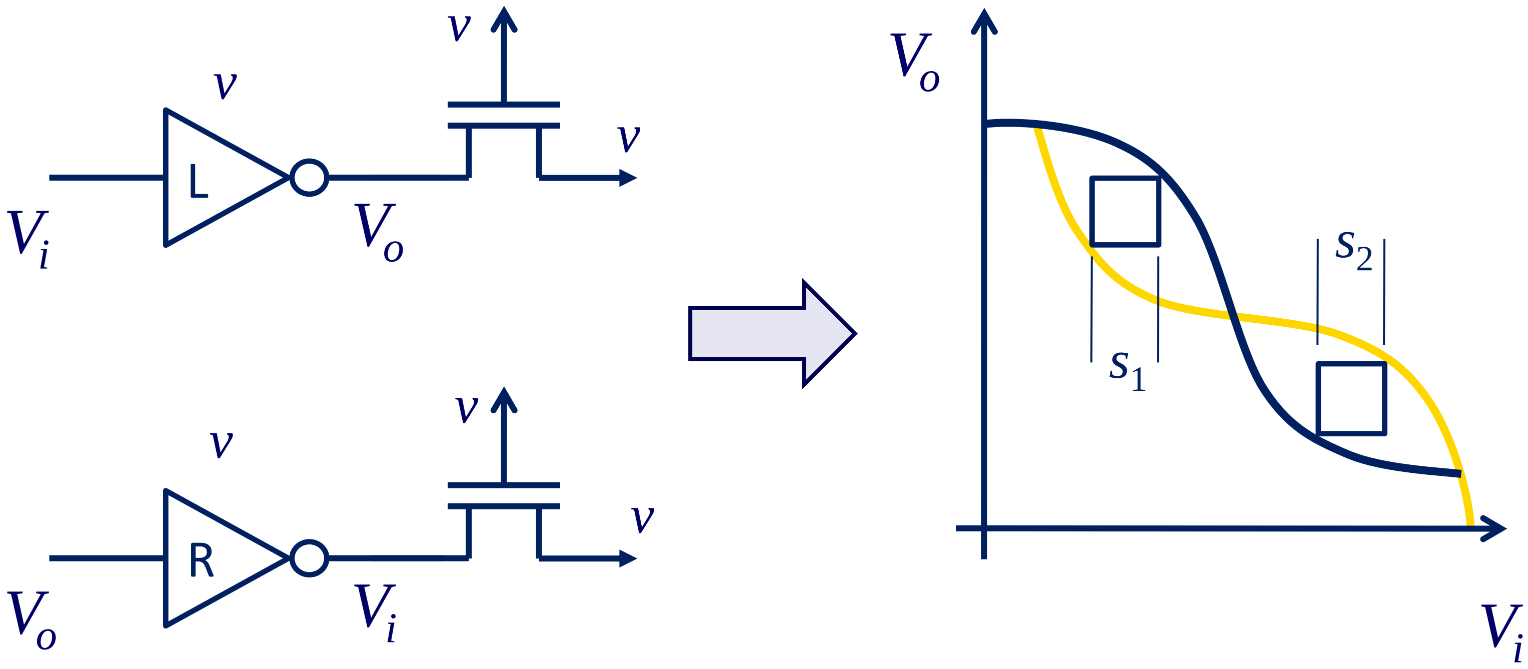
due to process variations is negligible (2-5%) for v in $[0.3, 1.0]$

[Kumar(Thesis)'08, Kumar-Rabaey-Ramchandran'09]

v	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$\log_{10}(r_s(v))$	-15.27	-15.29	-15.31	-15.34	-15.38	-15.43	-15.48	-15.54

Soft-error rate at $v = 0.3\text{V}$ is about 1.9 times of Soft-error rate at $v = 1.0\text{V}$

Parametric failures => read upset



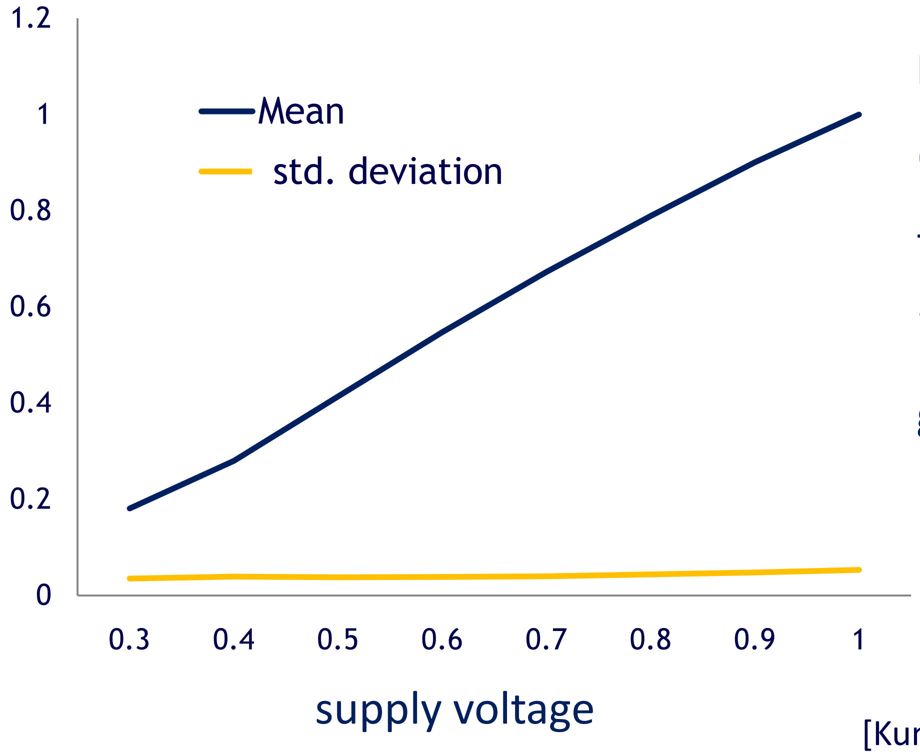
◊ Read (static) noise margin is defined as $RNM = \min(s_1, s_2)$

◊ By convention, if S_1 or S_2 is absent, $RNM < 0$

◊ $RNM \leq 0$ indicates a read-upset event

◊ It has been observed that $RNM \sim N(\mu_r, \sigma_r^2)$ [Nassif-Agarwal'06]

Parametric failures => $p_r(v)$



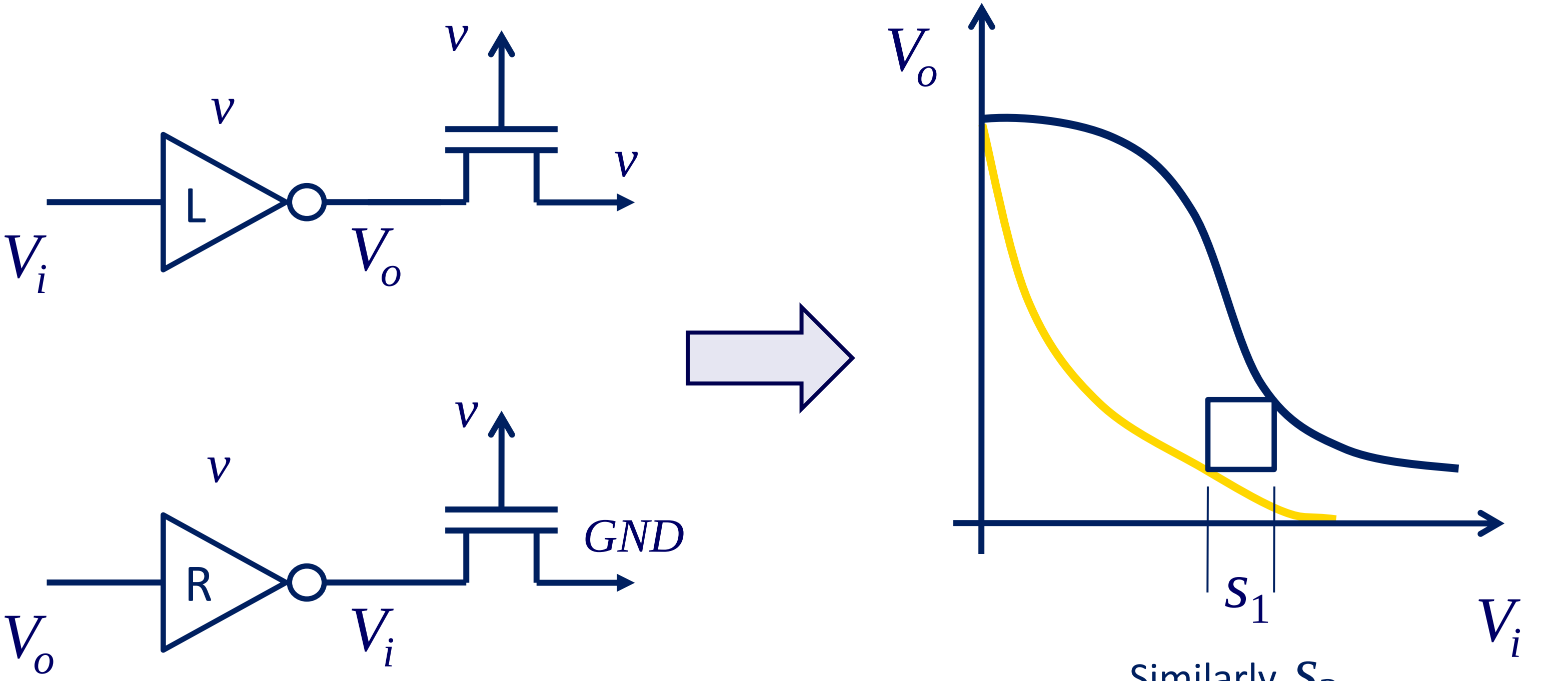
Read noise margin: Observed RNM (normalized) statistics (mean/var) as a function of supply voltage are illustrated. Read upset probability is given by,

$$p_r(v) = Q\left(\frac{\mu_r(v)}{\sigma_r(v)}\right)$$

[Kumar(Thesis)'08, Kumar-Rabaey-Ramchandran'09]

v	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$\log_{10}(p_r(v))$	-7.0	-12.3	-27.3	-45.8	-62.7	-72.9	-78.7	-78.9

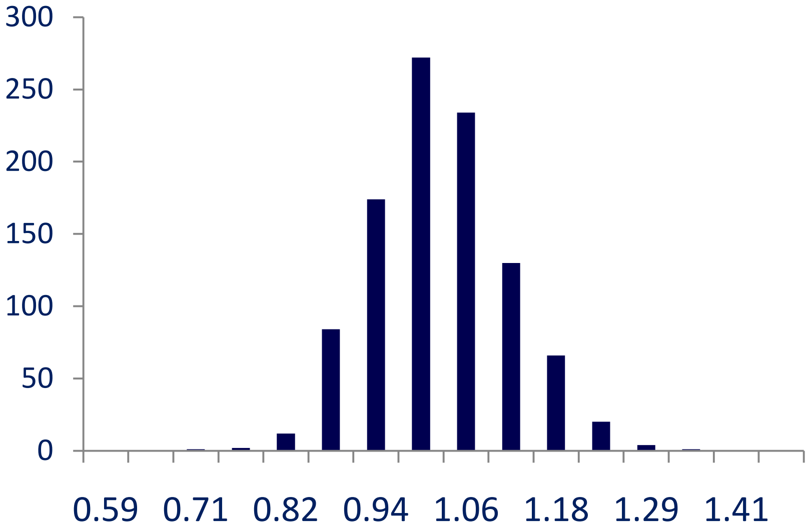
Parametric failures => write failure



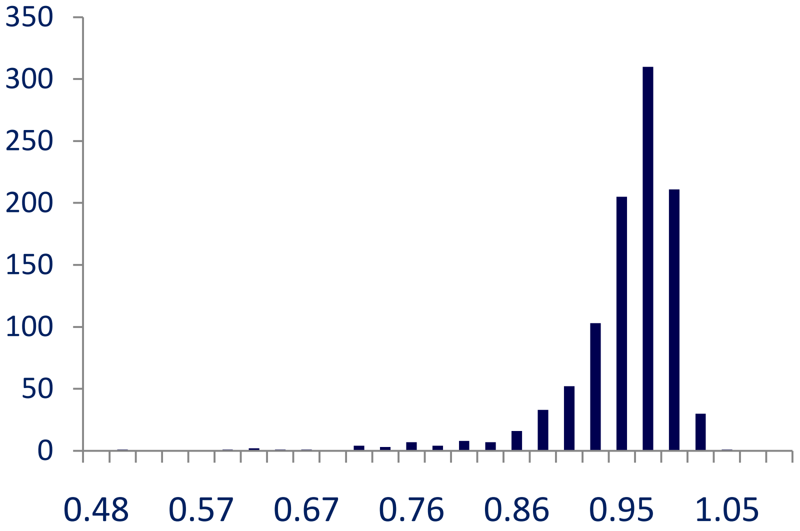
Similarly, S_2

- Write noise margin is defined as $WNM = \min(s_1, s_2)$
- $WNM \leq 0$ indicates a write-failure event
- Distribution of WNM is more complicated to estimate

Parametric failures => *WNM* distributions



$v = 0.8\text{V}$



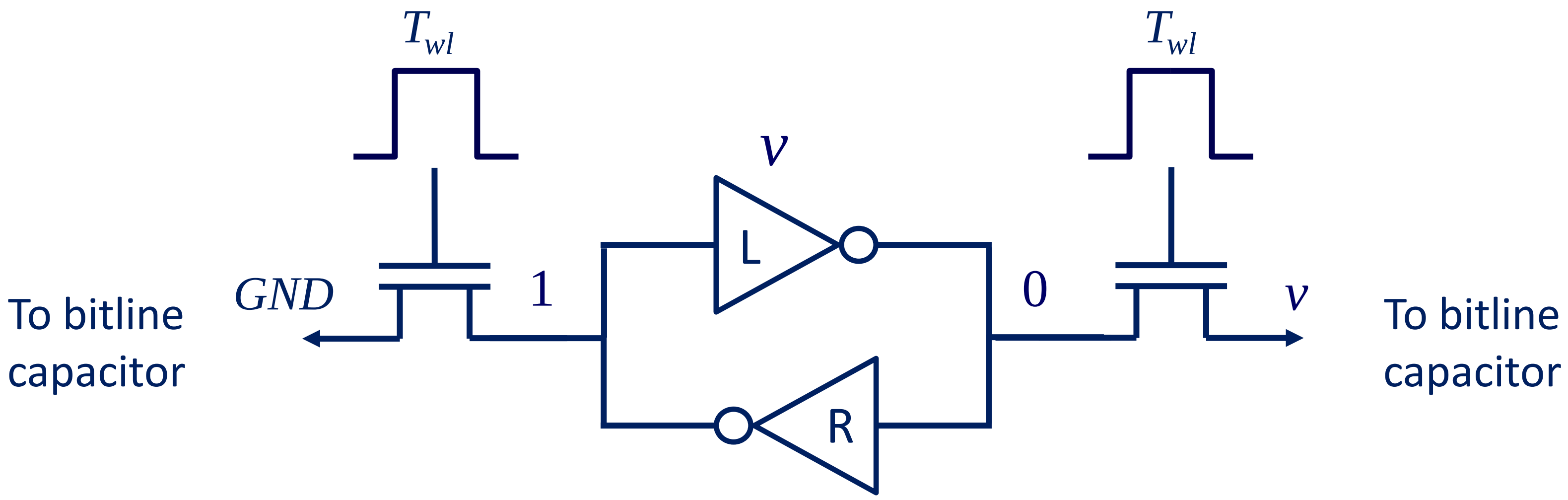
$v = 0.3\text{V}$

- ◊ WNM distribution is Gaussian at high-voltages
- ◊ Distribution head is estimated by exponential fit to the CDF $F_w(v)$

v	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$\log_{10}(p_w(v))$	-5.72	-6.20	-6.56	-11.09	…	…	…	…

[Kumar(Thesis)’08, Kumar-Rabaey-Ramchandran’09]

Parametric failures => write-time failures



◊ If T_{wl} is not large enough, then bit will not be written in successfully

◊ Due to process-variations, T_{wl} exhibits a distribution

◊ Need a method to estimate this distribution’s tail

CDF tail: extreme value theory

Use some results from Extreme Value Theory [Balkema-De Haan'74]

$$R_t(x) = P(X > x + t \mid X > t)$$

$$=> R_t(x)\, P(X > t) = P(X > x + t)$$

Result: If $R_t(x)$ converges for large t, then the limit is exponential

Remarks:

◊ Observe that exponential distributions satisfy this property for all $t > 0$

◊ It is hard to determine where the tail begins, and whether the distribution decay will change its behavior

◊ This technique is a good thumb-rule, in the absence of huge number of trials

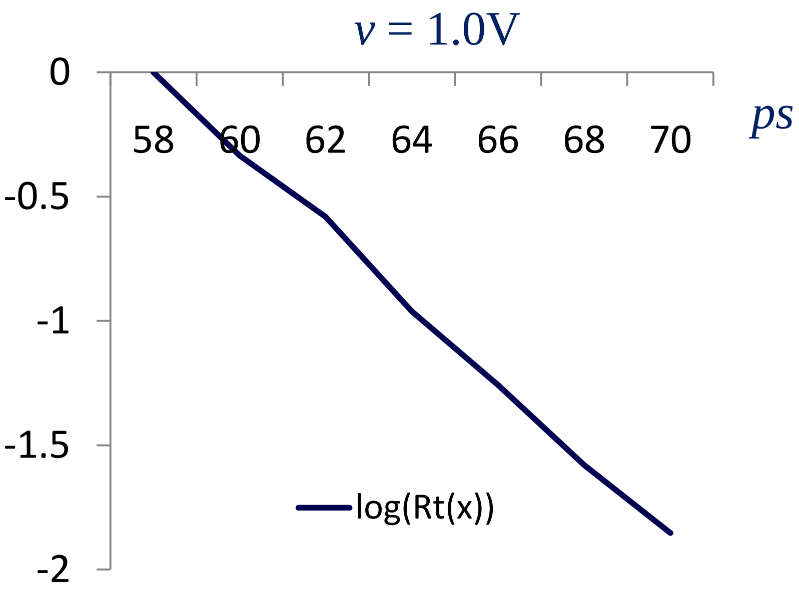
Write-time estimation

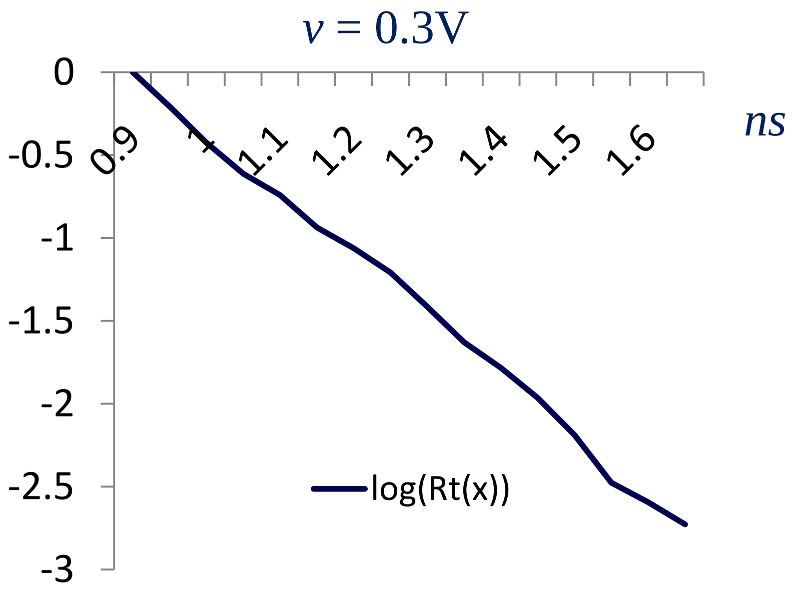
◊ From 2000 Monte Carlo trials, the 90% point (quantile) is chosen as t

◊ The residual probability function is plotted to check exponential behavior, if any

◊ Extrapolation used if exponential behavior is found

Remark: For all voltages, the exponential nature of decay was observed





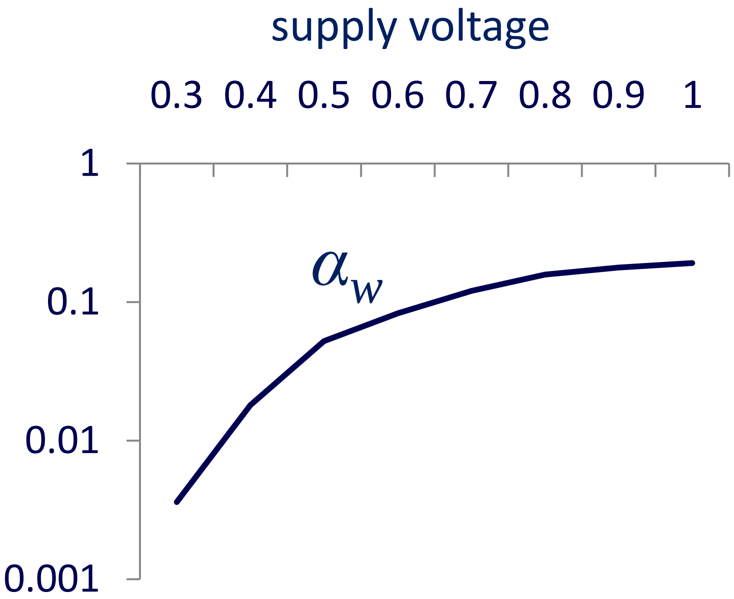
Write-time model

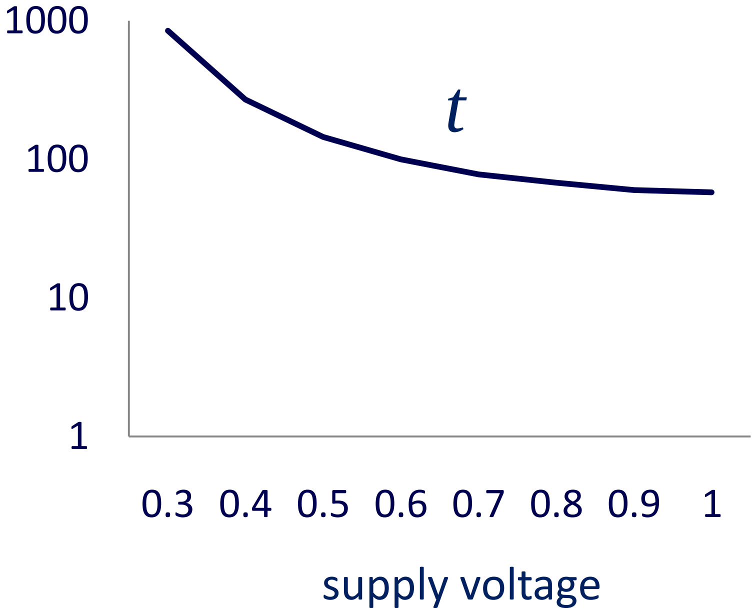
Using extreme value theory, the final write-time model is as follows:

$$P(T_{wl} > x + t) = P(T_{wl} > t)\exp(-\alpha_w x)$$

t: $P(T_{wt} > t) = 0.1$ [Kumar(Thesis)'08]

Using 2000 Monte Carlo trials, t and α_w were estimated as follows



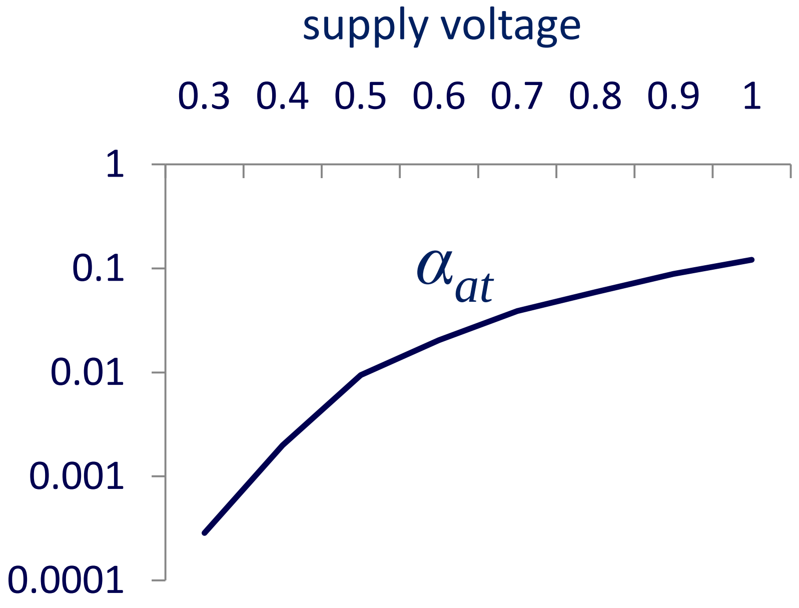


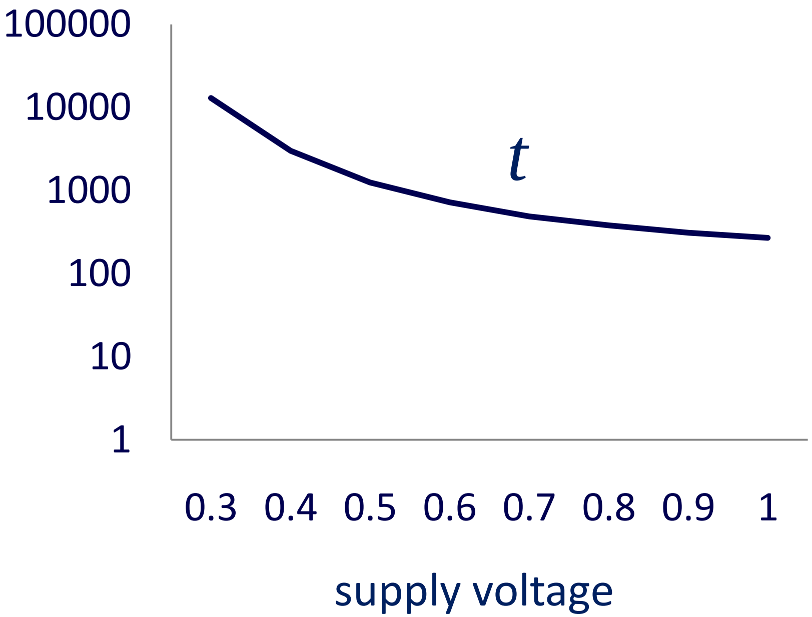
Access-time model

Similar to write-time, the following access-time model can be obtained:

$$P(T_{at} > x + t) = P(T_{at} > t)\exp(-\alpha_{at}\,x)$$

t: $P(T_{at} > t) = 0.1$ [Kumar(Thesis)'08]





Failure modeling summary

Soft-errors:

◊ Decreases exponentially with critical charge

◊ Critical charge evaluated using “noise”-current injection

◊ Monte-Carlo simulations to compute effect of process-variations

Parametric failures:

◊ Noise margin violations – Read, write, hold

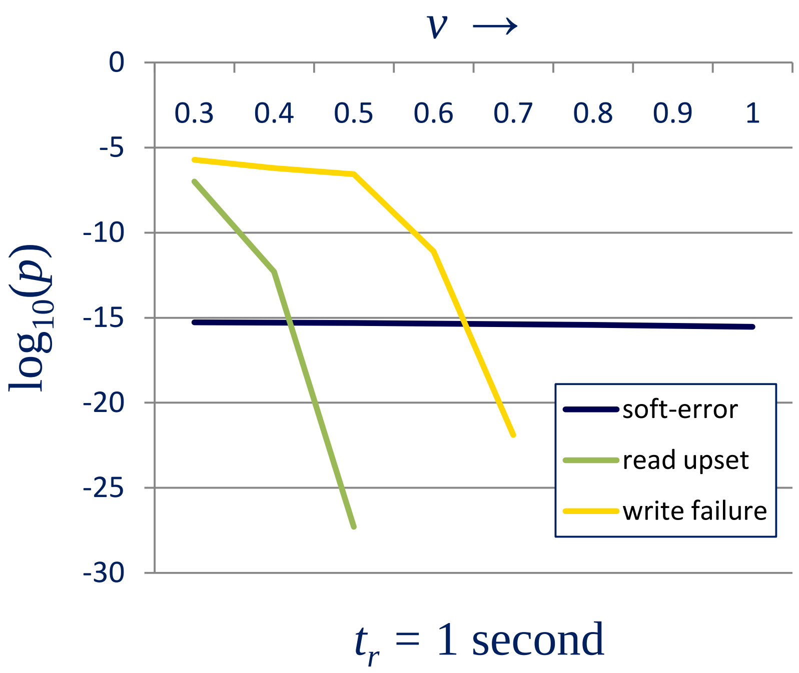
◊ Timing violations – write-time, access

◊ Estimated by Monte-Carlo and distribution-tail predictions

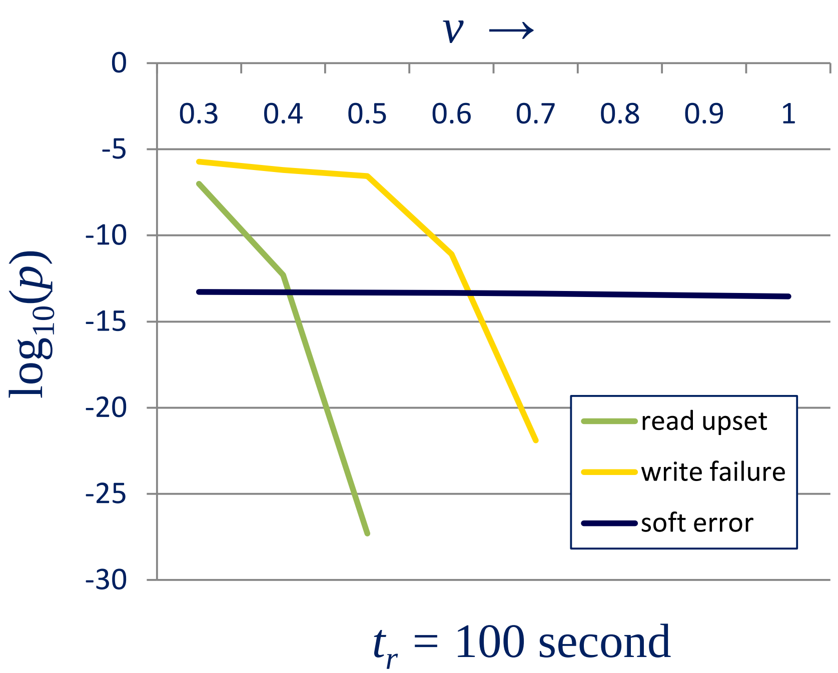
Supply noise:

◊ Tackled by 100mV margin

All failure probabilities combined



t_r = 1 second



t_r = 100 second

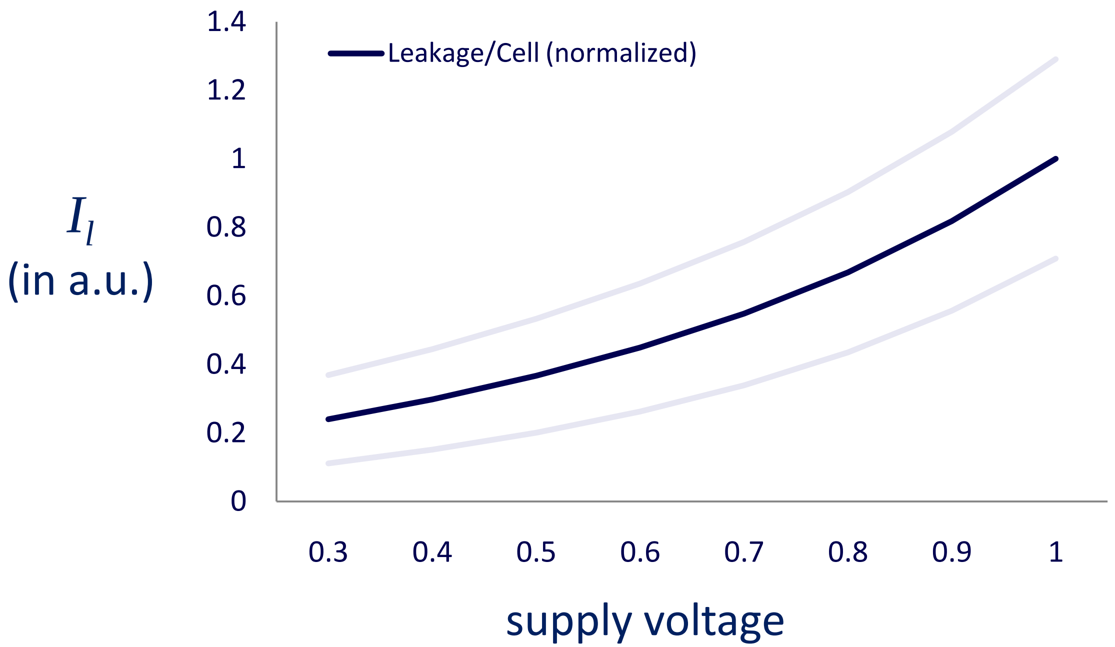
Observations

- At high voltages, soft-errors dominate the error probability
- At low voltages, parametric failures (notably write failures) take over
- At 0.2V, the SRAM cells were not writeable [Kumar(Thesis)'08]

Outline

- ◊ Introduction and contributions
- ◊ SRAM cell’s failure modeling (90nm CMOS)
- ◊ Leakage power optimization results (90nm CMOS)
- ◊ Conclusions

Leakage power dependence on supply



- This leakage power is obtained by averaging over 1000 trials, using Monte Carlo simulations
- The faded lines mark the (+/- 1σ) limits

Cost function and coding families

Cost function with $[n, k, d]$ code :

$$\mathscr{P}_b(v, t_r, ECC) = \frac{n}{k} P_l(v) + \frac{n}{k} \frac{4E_r + 3E_w + E_{ECC}}{t_r}$$

Target error-probability:

$n = 31$, $k = 26$, $d = 3$; $v = 1.0\text{V}$, t_0 = data-lifetime, sets

$$p_{target} = \binom{31}{2} (t_0 \cdot r_s(1.0))^2 = 1.40 \times 10^{-25} (t_0)^2$$

Coding families:

◊ Hamming : $d = 3$, $(n, k) = \{(31, 26), (63, 57), \ldots, (511, 502)\}$

◊ BCH codes : $d = 5, 7, \ldots, 15$; $n = \{63, 127, \ldots, 1023\}$

◊ $n < 1024$ for approx. spatial statistical independence in 256 by 256 SRAM blocks

Decoding error events

◊ For $[n, k, d]$ bounded distance decoding codes

◊ **Generalized decoding error event**: x errors and y erasures with probabilities $p_e(v)$ and $p_x(v)$, respectively, have the following error event [Forney'66, Lin-Costello'83]

$$2x + y > d - 1$$

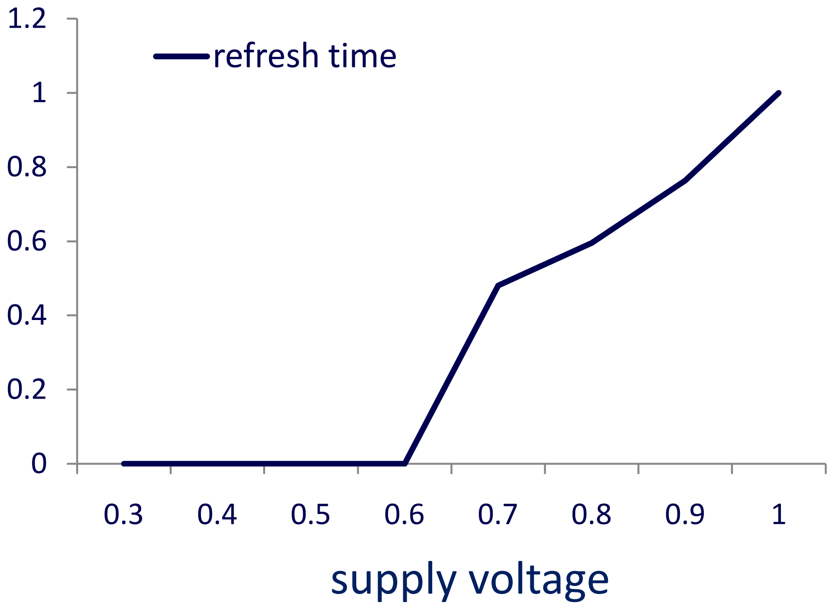
◊ **Specialized decoding error event:** z errors with probability $p_e(v)+p_x(v)$ has the following error event

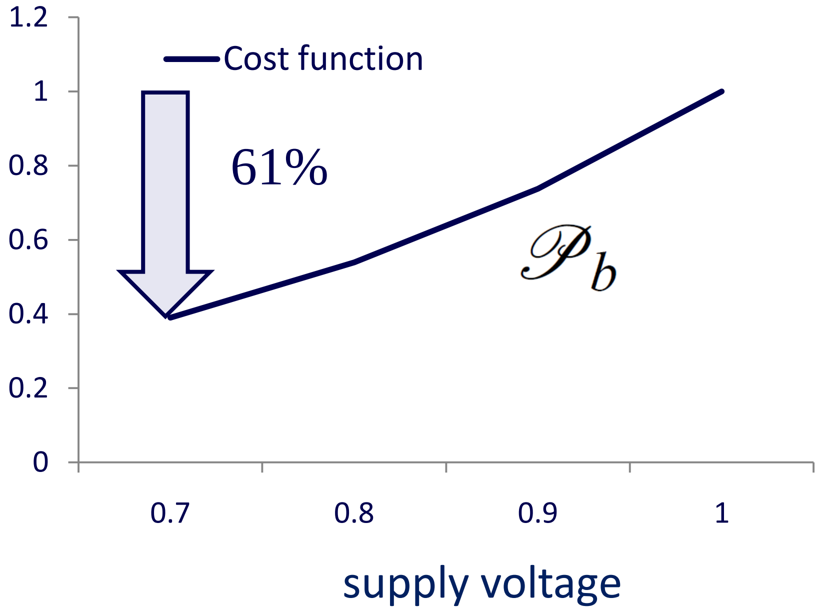
$$2z > d - 1$$

Effect of data-refresh (scrubbing)

Fix code to [31, 26, 3] Hamming code, and $t_0 = 1$ sec

Choose $(t_r\,, v)$ pairs to meet error-probability target



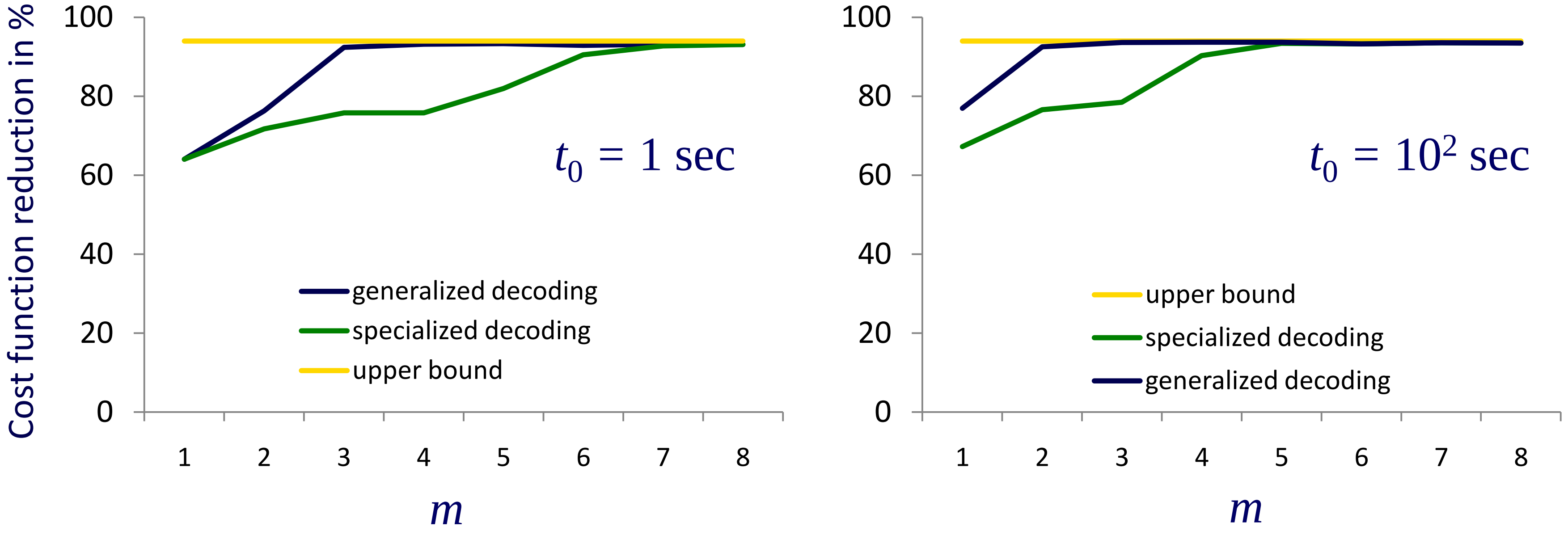


◊ Leakage power reduction with data-refresh is limited by parametric-failures

Leakage power and ECC trade-offs

For each ECC, Choose (t_r, v) pairs to meet the error-probability target

Find the optimum of cost function for each min. distance among ECC families



m = number of errors that can be corrected

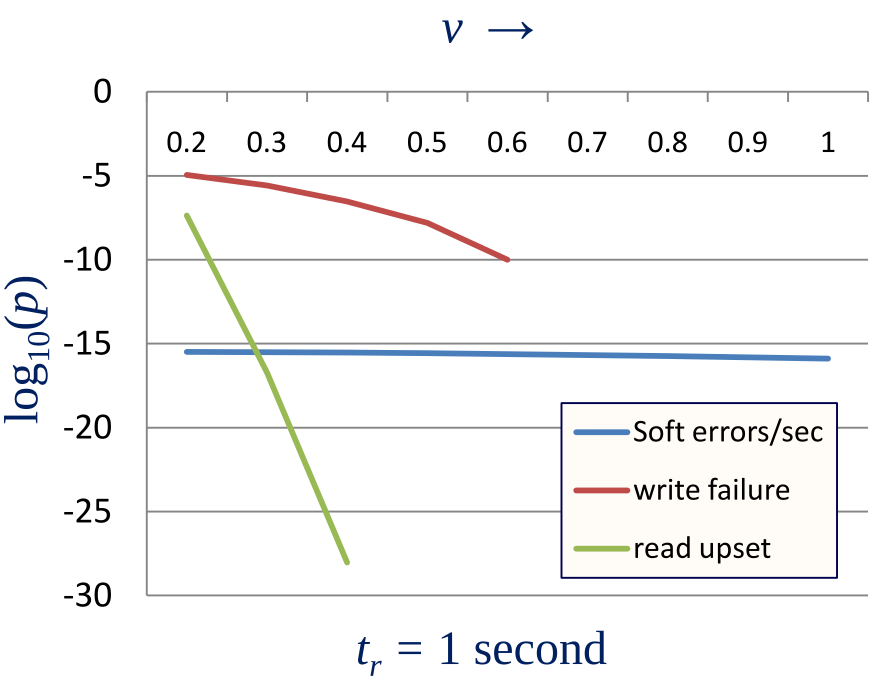
◊ Accounting for parametric failures as erasures leads to better power reduction

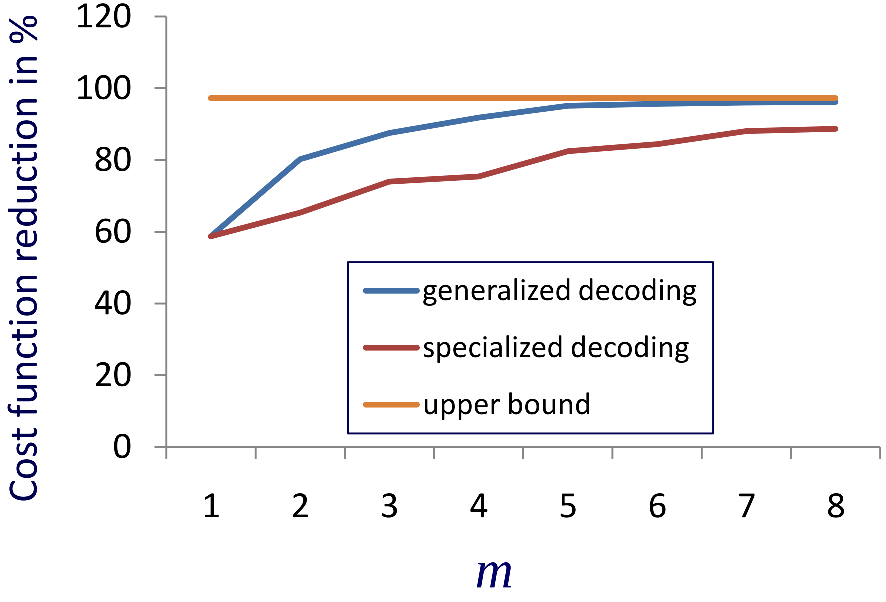
◊ The upper bound is 94% and 91% reduction is achieved by [127, 106, 7] BCH code

[Kumar(Thesis)'08, Kumar-Rabaey-Ramchandran'09]

Similar trade-offs for 65nm tech.

The same error-probability estimation and algebraic optimization routine can be run in sequence to obtain results for CMOS 65nm (low-leakage) technology





Full set of comparisons can be found in my thesis [Kumar(Thesis)'08]

Outline

◊ Introduction and contributions

◊ SRAM cell’s failure modeling (90nm CMOS)

◊ Leakage power optimization results (90nm CMOS)

◊ Conclusions

Conclusions

- Refresh and error-correction prevent reliability degradation in SRAM bits with voltage scaling. The leakage power reduction estimates are around 91% for [127, 106, 7] BCH code, and 61% for Hamming codes with $n < 512$
- Refresh (scrubbing) can also be used for improving reliability at fixed power
- Sieving spatially random soft-errors and spatially fixed parametric errors leads to possible leakage power reduction with lower complexity codes

Standby SRAM remarks

◊ In the special case of standby storage (for sensors), read and write based parametric failures can be ignored, thereby resulting in standby leakage-power optimization

◊ Using this approach, experimental chips were fabricated and tested in the 90nm CMOS technology by Huifang Qin (courtesy STMicroelectronics)

Related work on standby SRAM:

1. *"Fundamental data retention limits in SRAM standby – Experimental results,"* **A. Kumar,** H. Qin, P. Ishwar, J. Rabaey, and K. Ramchandran, ISQED, San Jose, CA, USA, Mar 2008.
2. *"Error-Tolerant SRAM Design for Ultra-Low Power Standby Operation,"* H. Qin, **A. Kumar**, P. Ishwar, J. Rabaey, and K. Ramchandran, ISQED, San Jose, CA, USA, Mar 2008.
3. *"Fundamental redundancy versus power trade-off in standby SRAM,"* **A. Kumar,** H. Qin, P. Ishwar, J. Rabaey, and K. Ramchandran, ICASSP 2007, Honolulu, Hawaii, USA, Apr 2007.

Acknowledgments

- **Advisors:** Dr. Kannan Ramchandran and Dr. Jan Rabaey
- **Intel SRAM design team** (Dr. T. M. Mak, Dr. M. Roncken, Dr. R. Mathur, Dr. M. Spica, and Dr. M. Zhang)
- **Research Groups:** BASiCS, pJoules, and Berkeley Wireless Research Center (BWRC) at Univ. of California, Berkeley
- **Technology Access:** ST Microelectronics (model files)
- **Funds:** Gigascale silicon research center and National science foundation