Self-Repairing and Self-Calibration: A Design/Test Strategy for Nano-scale CMOS

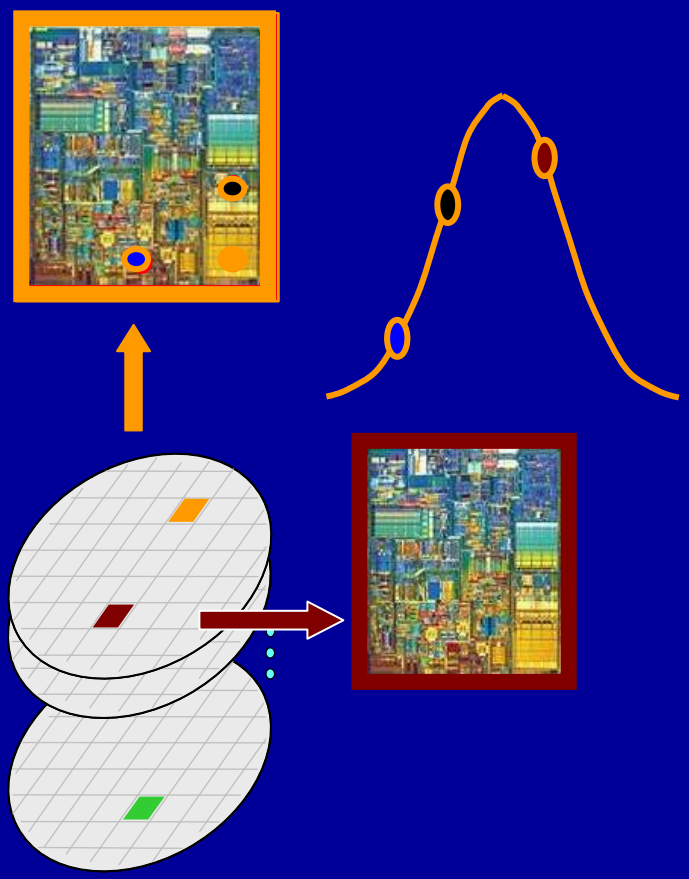
Kaushik Roy

S. Mukhopadhyay, H. Mahmoodi, A. Raychowdhury, Chris Kim, S. Ghosh, K. Kang

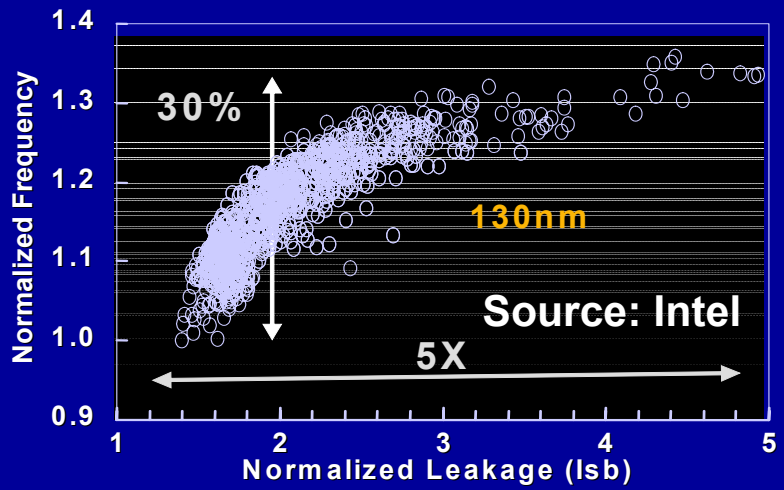
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN

PURDUE
UNIVERSITY

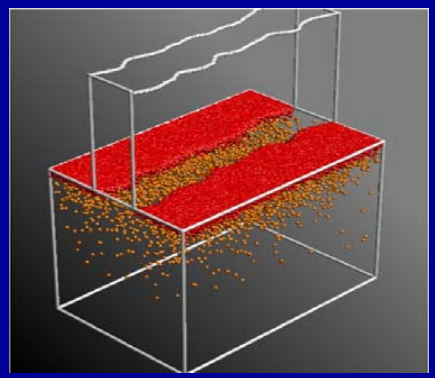
Process Variation in Nano-scale Transistors



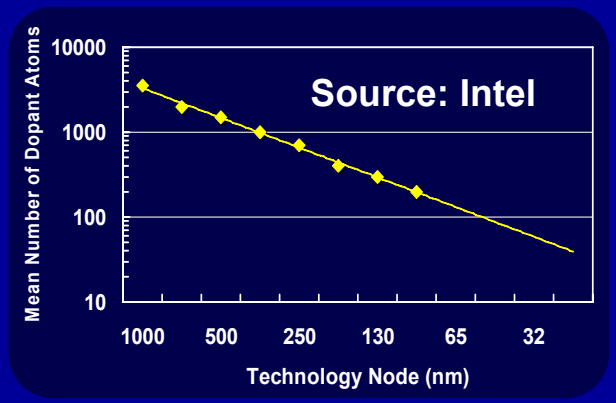
Inter and Intra-die Variations



Delay and Leakage Spread



Random dopant fluctuation



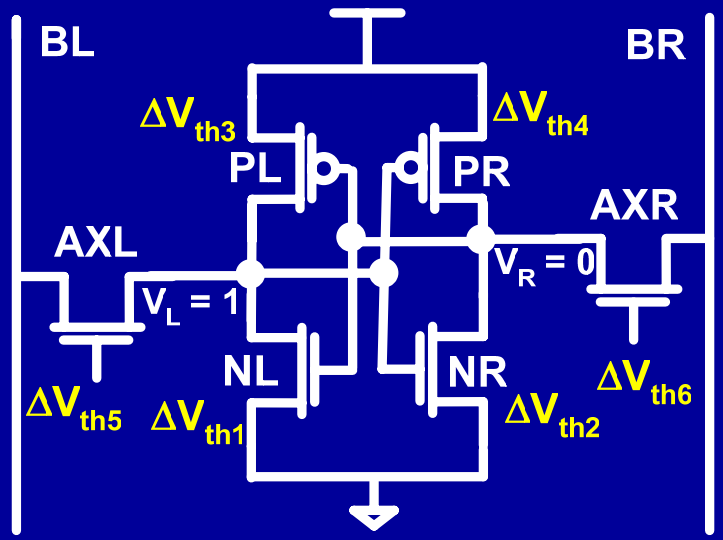
- Device parameters are no longer deterministic

Process Variations: Failures, Test, Self-Calibration, and Self Repair

- Memories (Caches and Register Files)
  - Failure analysis
  - Updated March Test for process induced failures
  - Process/Defect tolerant caches
  - Self-Repairing SRAM's
  - Leakage and delay sensors for self-repair
- Logic
  - Failure analysis in pipelines and robust pipeline design
  - Delay sensor to measure critical path delays and integrated test generation for robust segment delay coverage
- Updated scan chain logic
  - FLH and FLS (First level hold and scan flip-flops) for low-power (dynamic and leakage) and efficient delay testing
- Temperature control to prevent thermal runaway during burn-in

SRAM Memory Cell Failure

- **Process variation:**
  Device miss-match → Cell failure
- **Primary source:**
  Random dopant fluctuation
  → V_{th} miss-match
- **Results in three types of failures**



AF: Access time failure

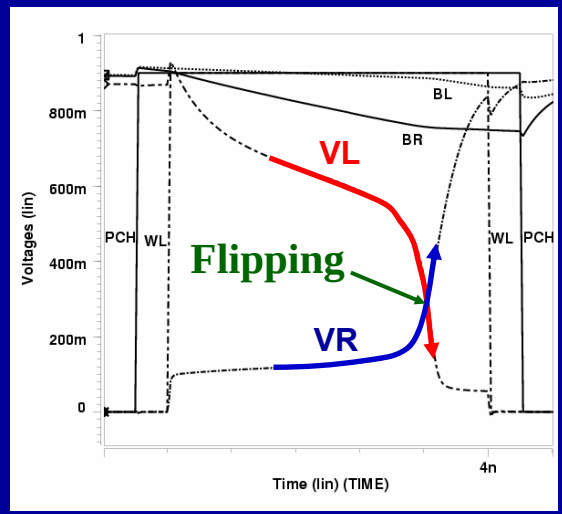
RF: Read failure

WF: Write failure

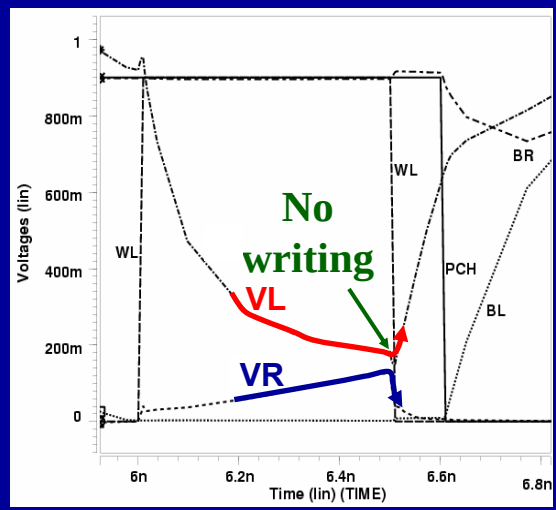
HF: Hold failure

$T_{ACCESS} > T_{MAX}$

AF: Access time failure

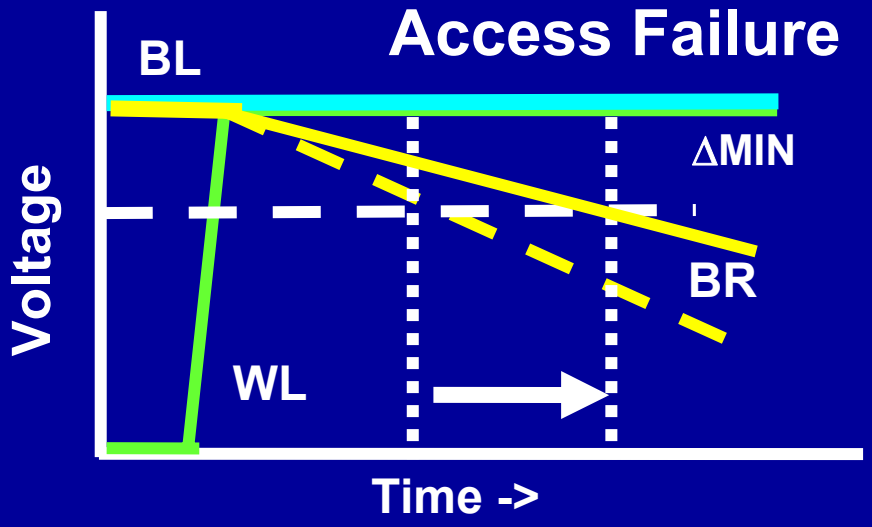


RF: Read failure

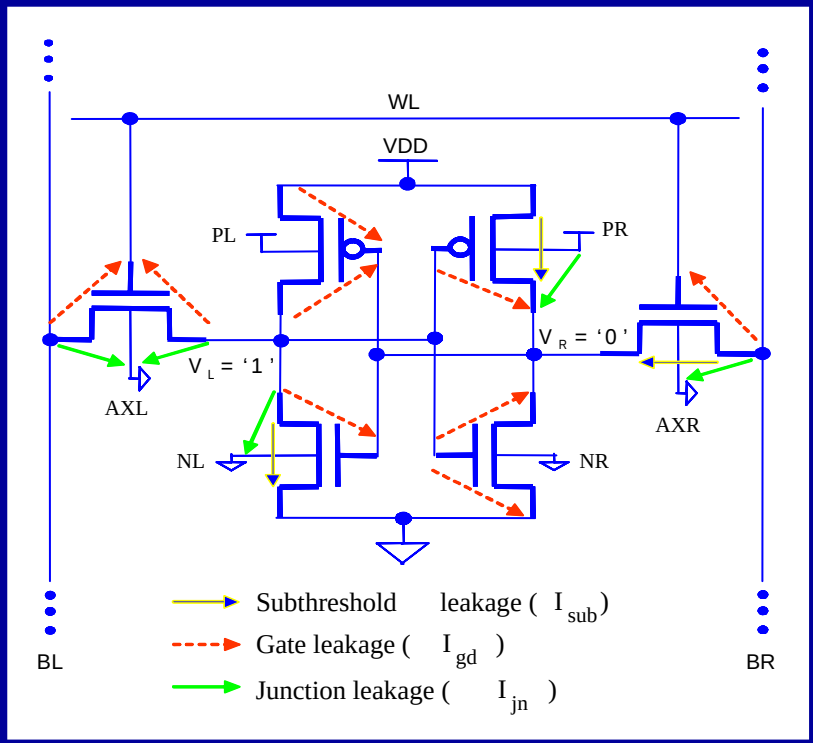


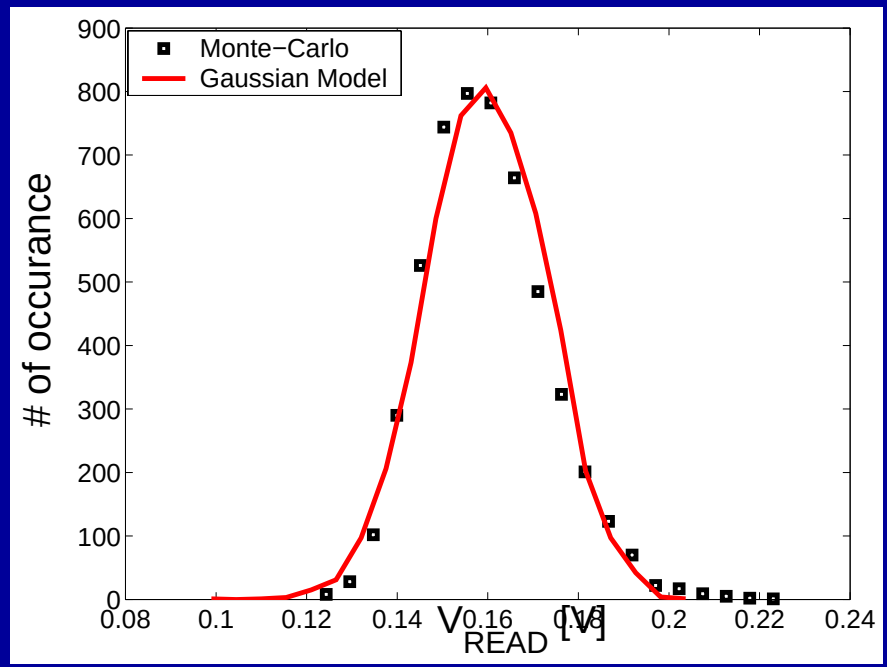
WF: Write failure

Mechanisms of Parametric Failures



Read Failure (RF)





Distribution of V_{READ}

$$P_{RF} = P\left(V_{READE} > V_{TRIPRD}\right)$$

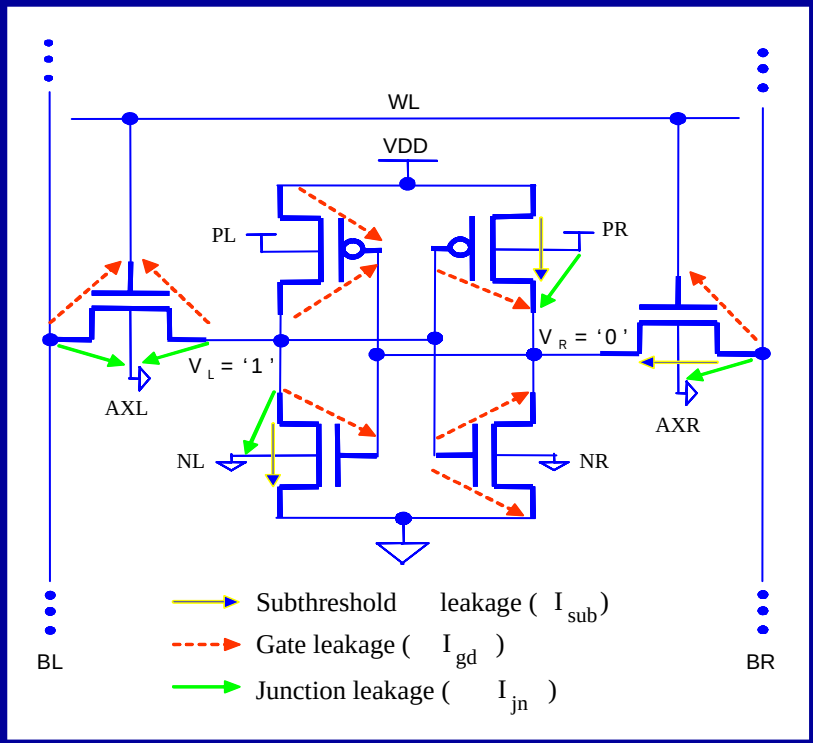
$I_{dsatAXR} = I_{dlinNR}$ → Solve for V_{READ}

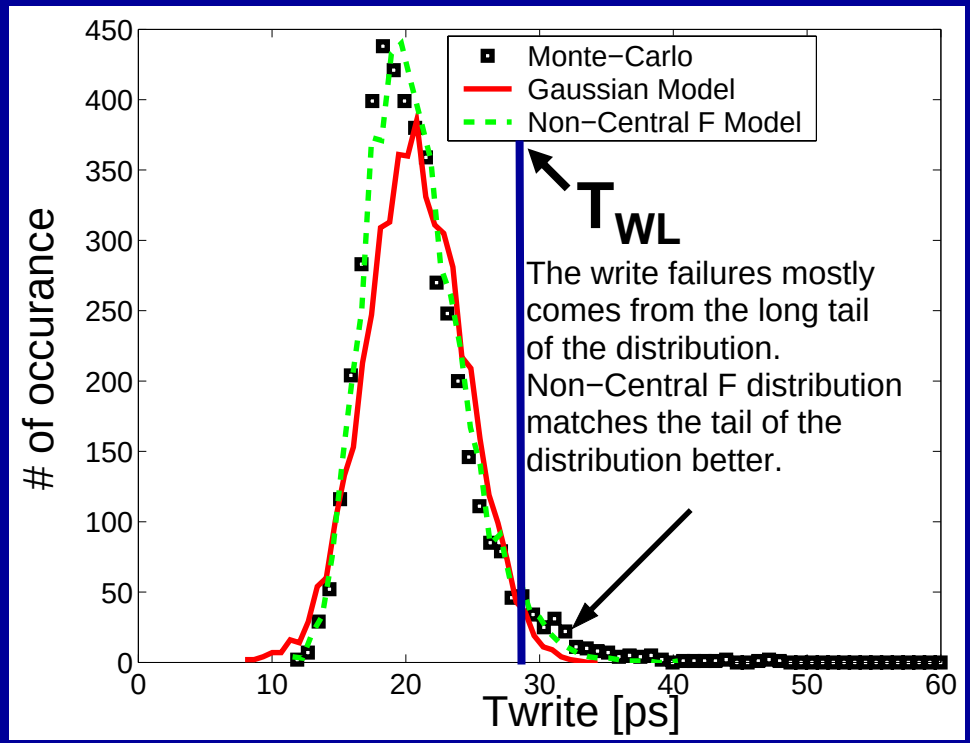
$I_{dsatNL}(V_{TRIPRD}, V_{TRIPRD}, gnd) \approx I_{dsatPL}(V_{TRIPRD}, V_{TRIPRD}, V_{DD})$ → Solve for V_{TRIPRD}

$$P_{RF} = P\left[Z \equiv \left(V_{READ} - V_{TRIPRD}\right) > 0\right] = 1 - F_Z(0)$$

where, $\eta_Z = \eta_{V_{READ}} - \eta_{V_{TRIP}}$ and $\sigma_Z^2 = \sigma_{V_{READ}}^2 + \sigma_{V_{TRIPRD}}^2$

Write Failure (WF)





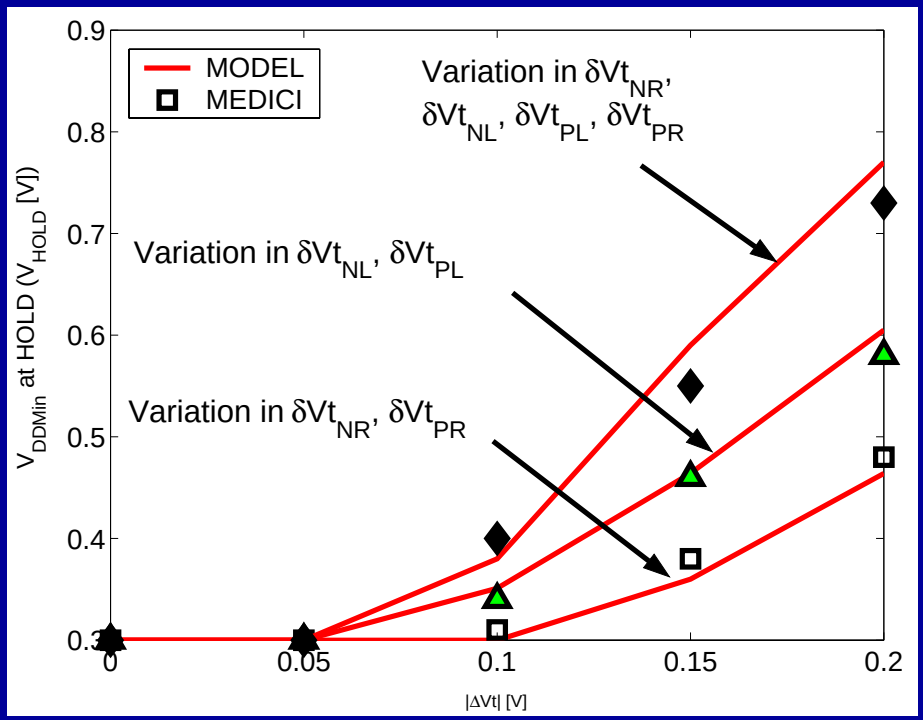
$$P_{WF} = P\left(T_{WRITE} > T_{WL}\right)$$

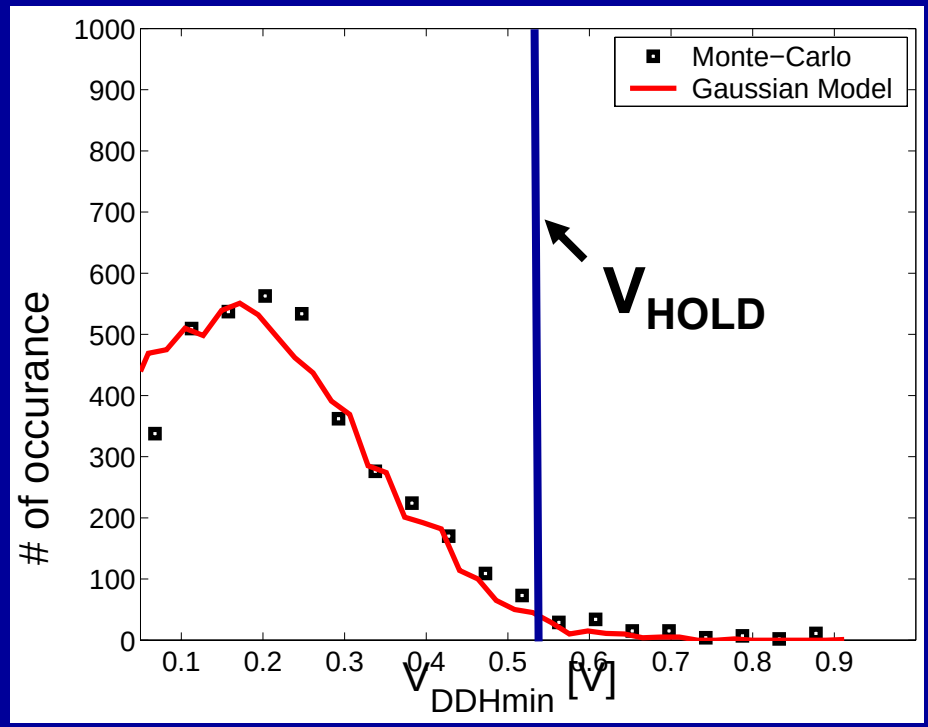
$$T_{WRITE} = \begin{cases} \int_{V_{DD}}^{V_{TRIP}} \frac{C_L(V_L)dV_L}{I_{in(L)}(V_L) - I_{out(L)}(V_L)}; & if\ (V_{WR} < V_{TRIPWR}) \\ \infty; & if\ (V_{WR} \geq V_{TRIPWR}) \end{cases}$$

$$P_{WF} = \int_{t_{WR}=T_{WL}}^{\infty} f_{WR}(t_{WR})d(t_{WR}) = 1 - F_{WR}(T_{WL})$$

$I_{in(L)}$ = current into L $\approx I_{dsPL}$, $I_{out(L)}$ = current out of L $\approx I_{dsAXL}$

Hold Failure (HF)





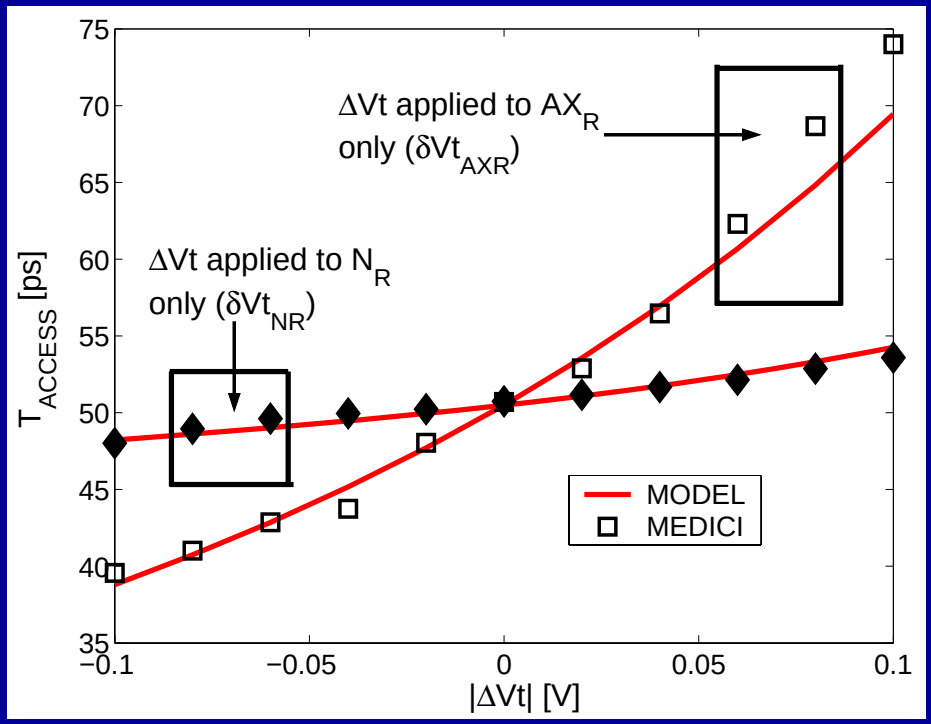
Variation of and Distribution of V_{DDHmin}

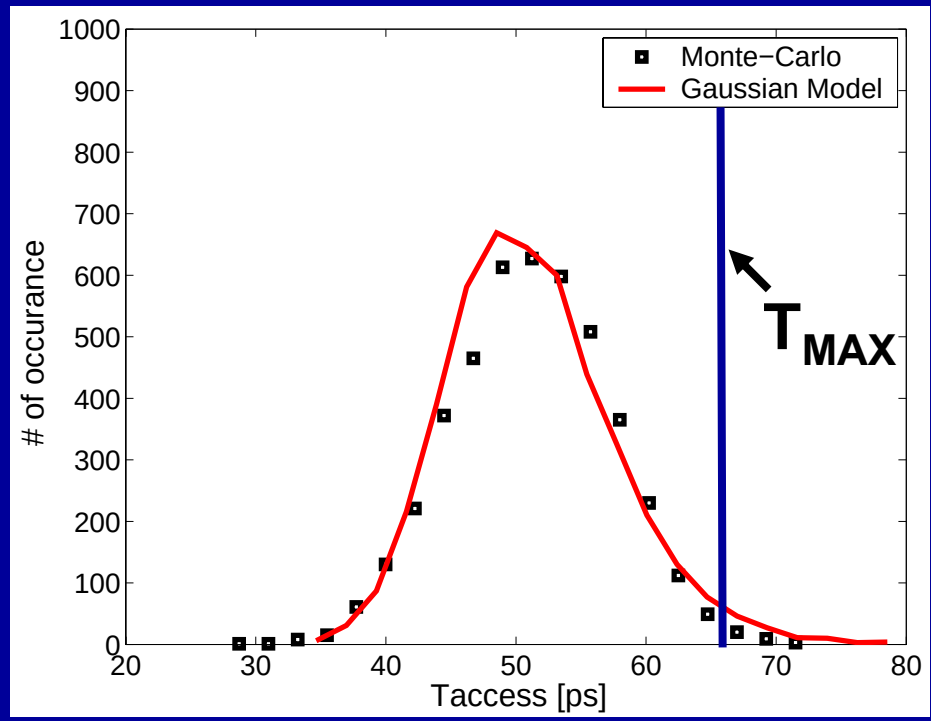
$$P_{HF} = P\left(V_{DDHmin} > V_{HOLD}\right)$$

$$V_L\left(V_{DDHmin}, \delta Vt_{PL}, \delta Vt_{NL}\right) = V_{TRIP}\left(V_{DDHmin}, \delta Vt_{PR}, \delta Vt_{NR}\right) \Rightarrow \text{Solve for } V_{DDHmin}$$

$$P_{HF} = \int_{V_{HOLD}}^{\infty} f_{VDDHmin}(v_{DDHmin})\,d(v_{DDHmin}) = 1 - F_{VDDHmin}(V_{HOLD})$$

Access Time Failure (AF)





Variation and distribution of T_{ACCESS} with variation in δVt

$$P_{AF} = P(T_{ACCESS} > T_{MAX})$$

$$T_{ACCESS} = \frac{C_{BR} C_{BL} \Delta_{MIN}}{C_{BL} I_{BR} - C_{BR} I_{BL}} = \frac{C_B \Delta_{MIN}}{I_{dsatAXR} - \sum_{i=1,..,N} I_{subAXL}(i)}$$

$$P_{AF} = \int_{t_{ACCESS}=T_{MAX}}^{\infty} f_{TACCESS}(t_{ACCESS}) d(t_{ACCESS}) = 1 - F_{TACCESS}(T_{MAX})$$

Basic Modeling Approach

- Estimation of the mean and the variance of a function several independent normal random variable

$$T_{ACCESS} = f(V_{t1}, V_{t2}, \ldots, V_{t6})$$
$$= f(\eta_{Vt1}, \eta_{Vt2}, \ldots, \eta_{Vt6}) + \sum_{i=1,..,6} (\partial f / \partial V_{ti})(V_{ti} - \eta_{Vti}) + \ldots.$$

  - Expand 'f' in Taylor series with respect to $Vt_1,\ldots,Vt_6$ around their mean and consider up to 2nd order terms.
  - Estimate the mean and the variance as:

$$Mean(T_{ACCESS}) = f(\eta_{Vt1}, \ldots, \eta_{Vt6}) + \frac{1}{2} \sum_{i=1,\ldots,6} (\partial^2 f / \partial V_{ti}^2) \sigma_{Vti}^2$$

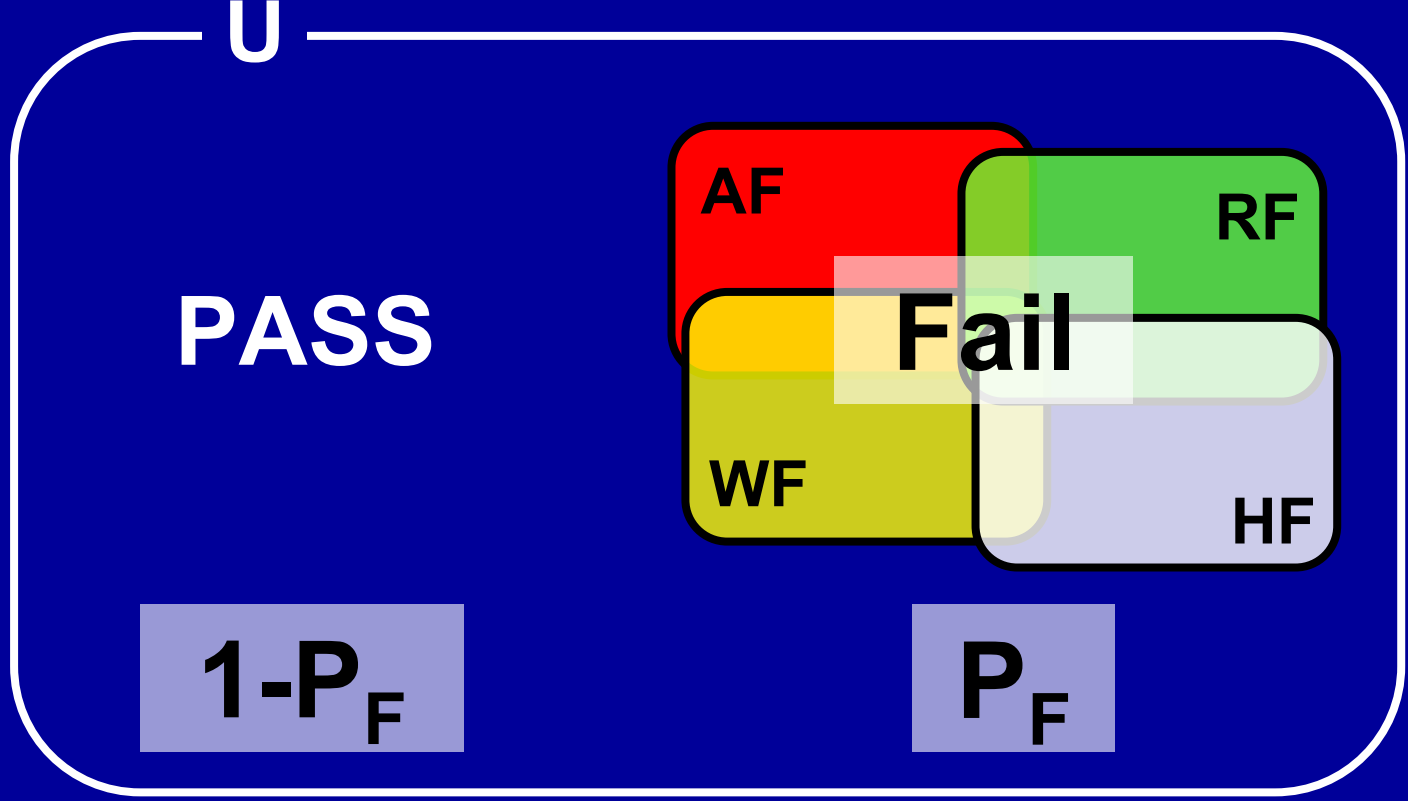
$$Variance(T_{ACCESS}) = \sum_{i=1,\ldots,6} (\partial f / \partial V_{ti})^2 \sigma_{Vti}^2$$

- Estimation of the probability distribution function of Y
  - Assume a normal pdf with the estimated mean and variance.
- The δVt of each transistors are assumed to be independent normal random variables.

$$\sigma_{\partial Vt} \alpha \left(1/\sqrt{LW}\right)$$

Estimation of Overall SRAM Cell Failure Probability (P_F)

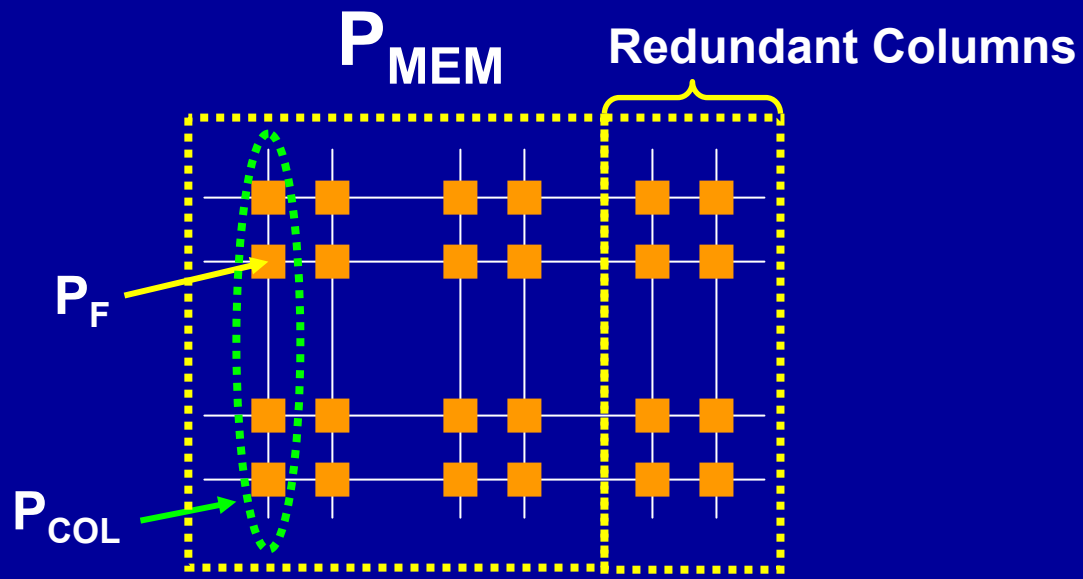
$$P_F = P[Fail] = P[A_F + R_F + W_F + H_F]$$



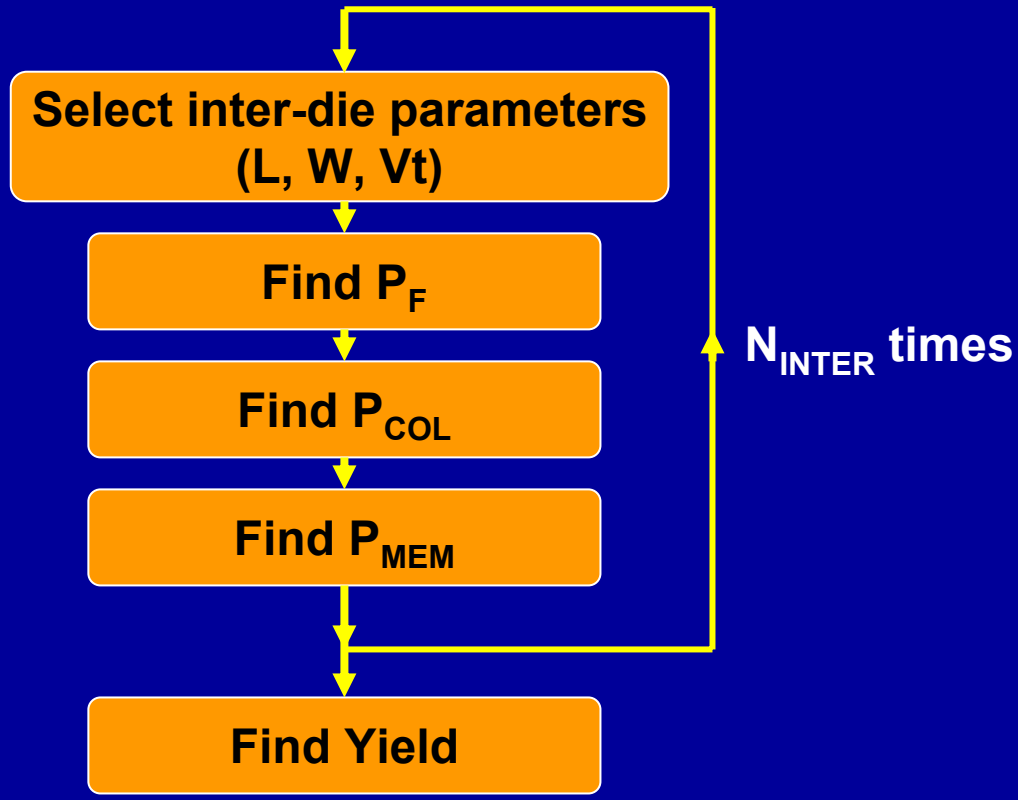
Estimation of Memory Failure Probability (P_{MEM})

- P_{COL}: Probability that any of the cells in a column fails
- P_{MEM}: Probability that more than N_{RC} (# of redundant columns) fail

$$P_{COL} = 1-(1-P_F)^N \quad \text{and} \quad P_{MEM} = \sum_{i=N_{RC}+1}^{N_{COL}} \binom{N_{COL}}{i} P_{COL}^{i} \left(1-P_{COL}\right)^{N_{COL}-i}$$



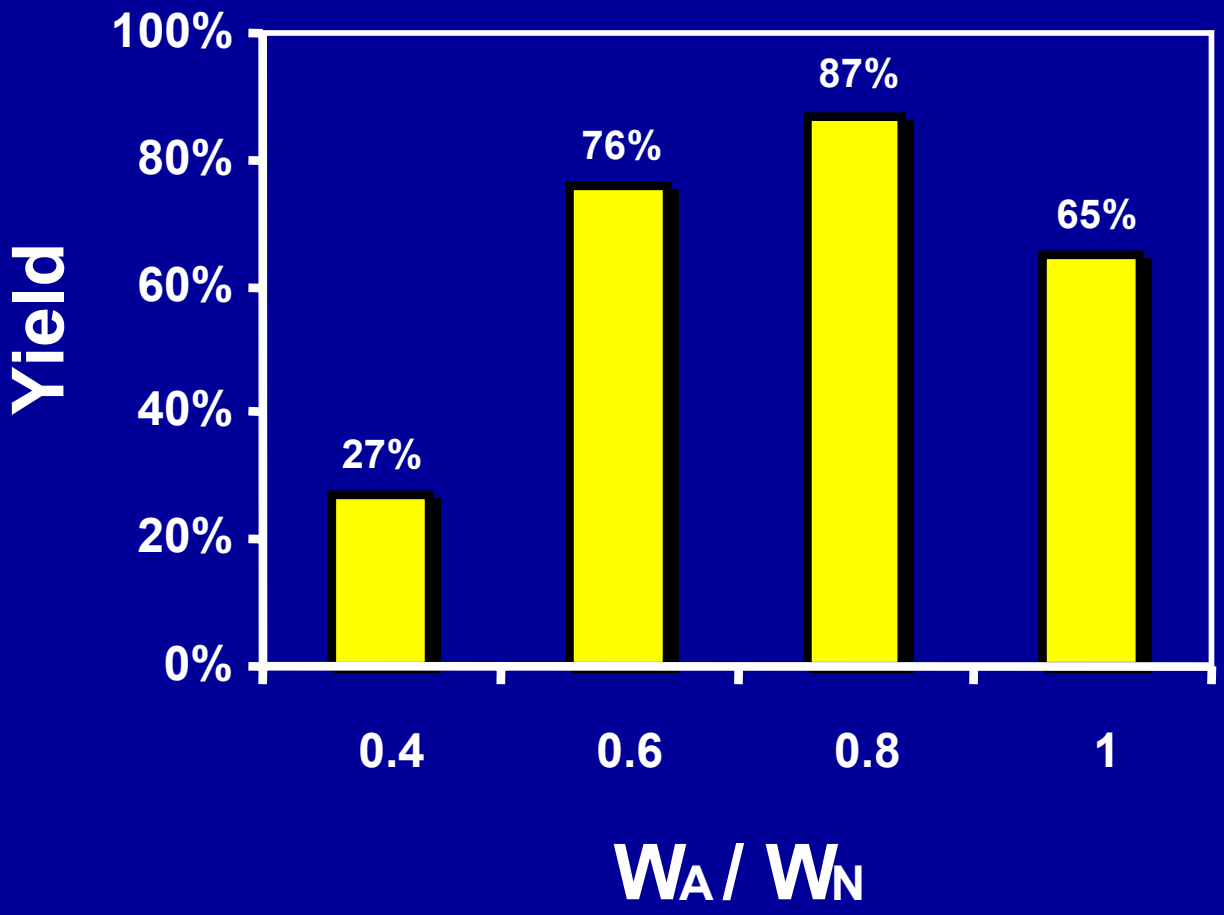
Estimation of Yield

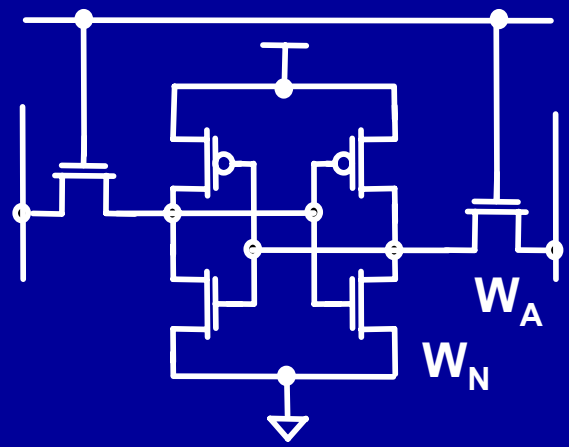


$$Yield = 1 - \left(\sum_{INTER} P_{MEM}\left(L_{INTER}, W_{INTER}, Vt_{INTER}\right) \Big/ N_{INTER} \right)$$

N_{INTER} : the total number of inter-die Monte-Carlo simulations (i.e. total number of chips)

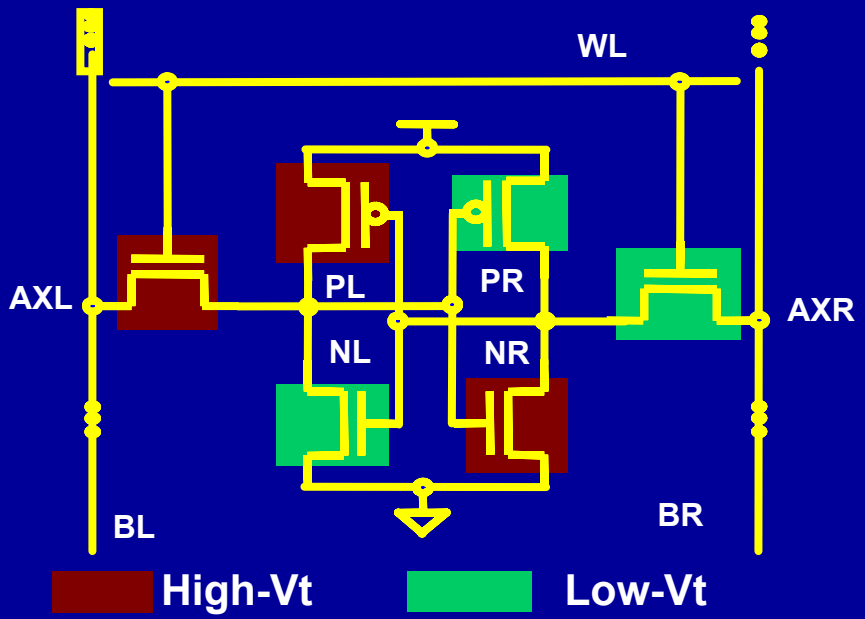
Snapshot: Transistor Sizing and Yield



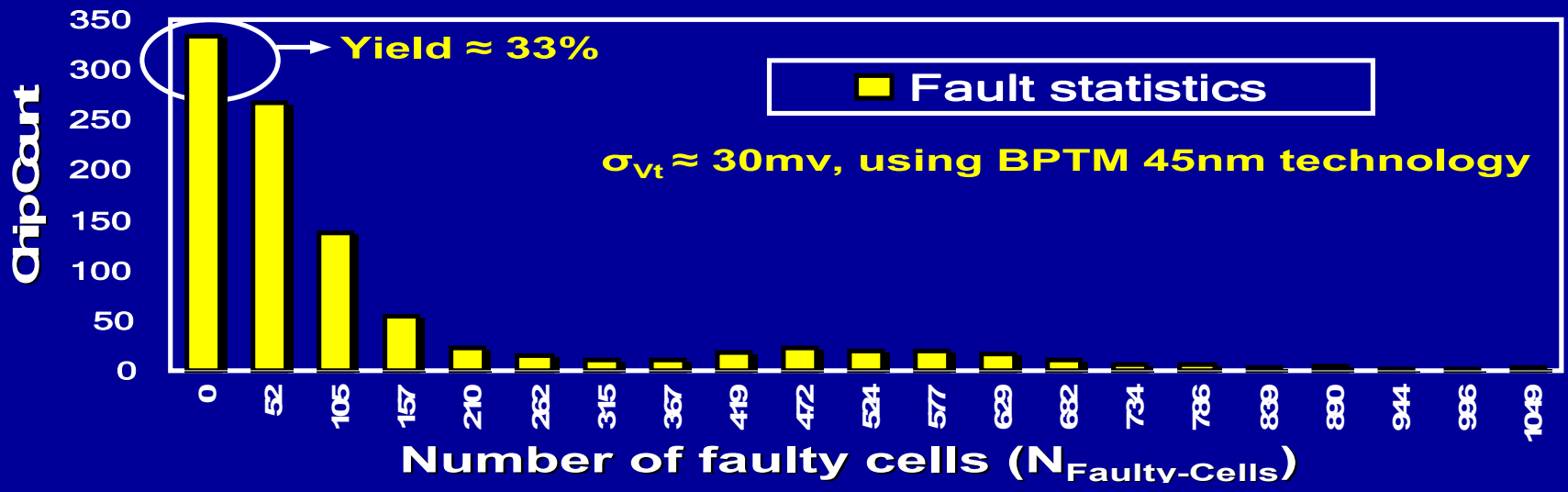


Proposed failure analysis assists SRAM designers to achieve maximum memory yield

Parametric Failures in SRAM Cell

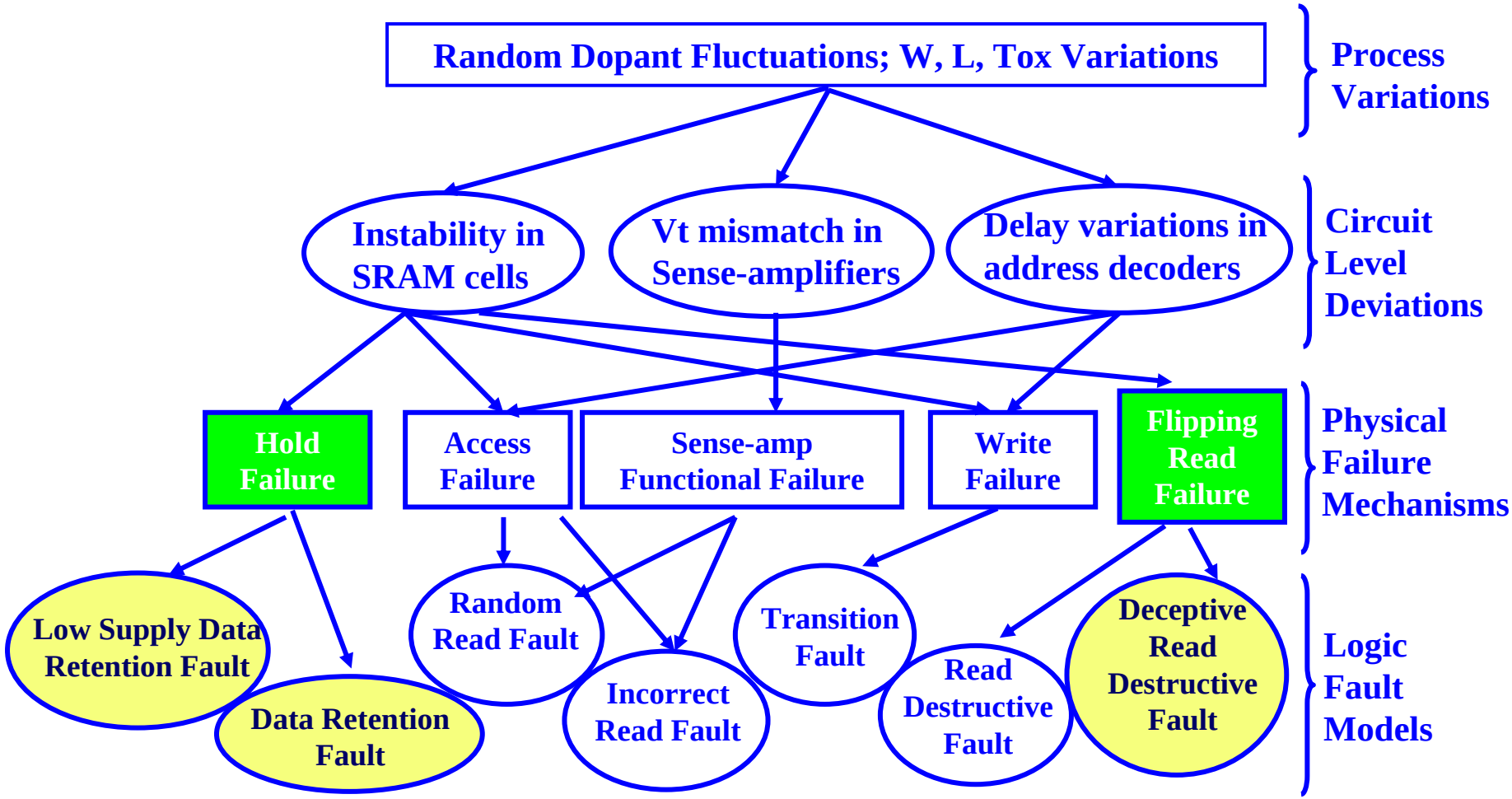


- Transistor mismatch => Parametric failures
  - Read failure
  - Write failure
  - Access failure
  - Hold failure



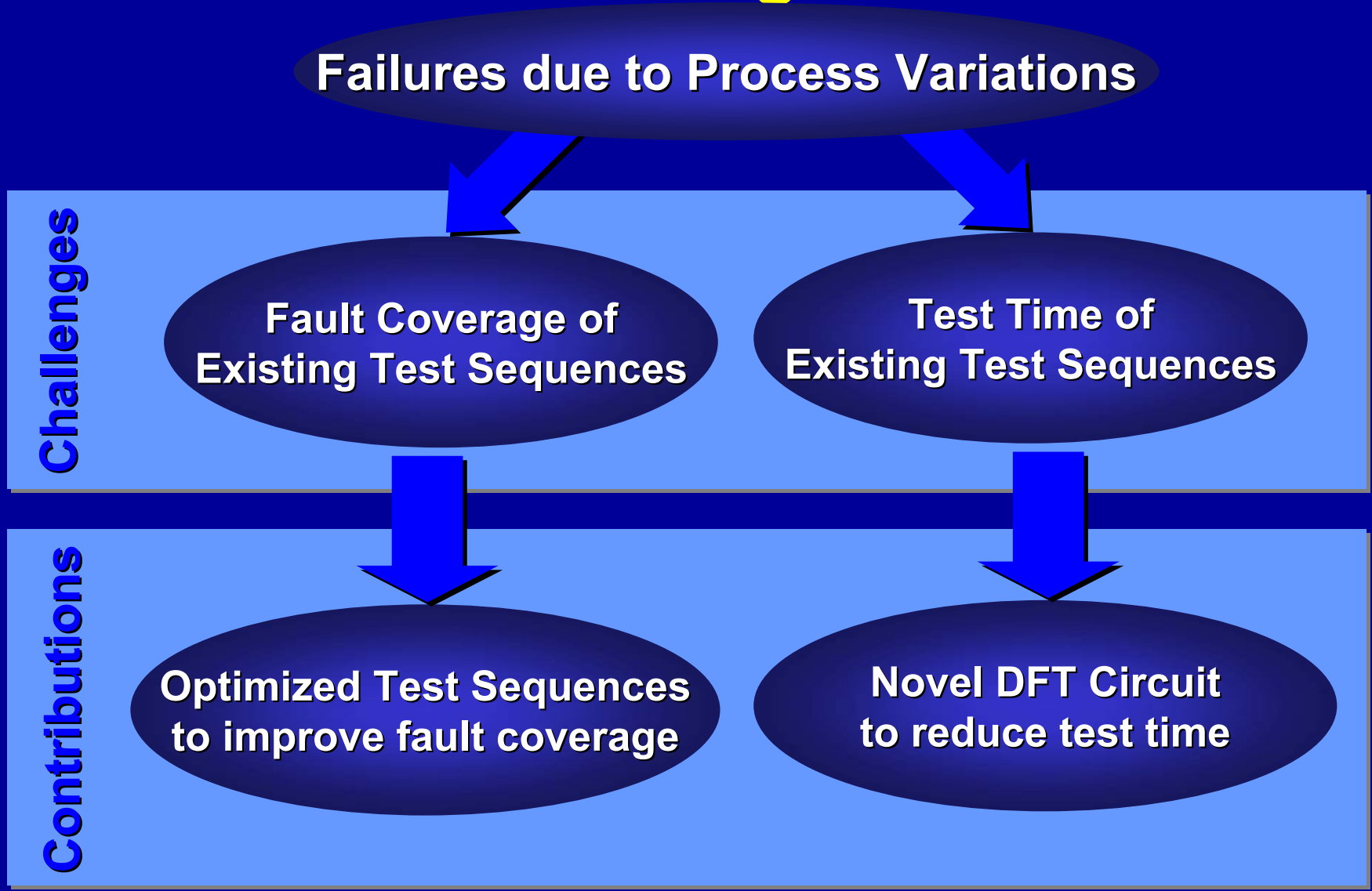
Parametric Failures => Yield degradation

SRAM Failure Mechanisms and Logic Fault Models



- Deceptive read destructive faults are overlooked in conventional test sequences
- Hold failures not detectable in conventional test sequences

Efficient Testing of SRAM*



* *IEEE VLSI Test Symposium*, May 2005

Test: Optimized March Test Sequence

1. Optimized March C-

($\Updownarrow$ W0) ($\Uparrow$ R0 W1) ($\Uparrow$ R1 W0) ($\Downarrow$ R0 W1) (HOLD)

($\Downarrow$ R1 R1 W0) (HOLD) ($\Updownarrow$ R0 R0)

+ Good fault coverage
- Test time increases

2. March Q

($\Downarrow$ W0) (HOLD) ($\Uparrow$ R0 W0 W1 R1)

(HOLD) ($\Uparrow$ R1 W1 W0 R0) ($\Downarrow$ R0)

+ Reduce the test time
+ Cover all the fault models induced by process variations
- Not able to detect Transition Coupling Fault and Address Decoder Fault

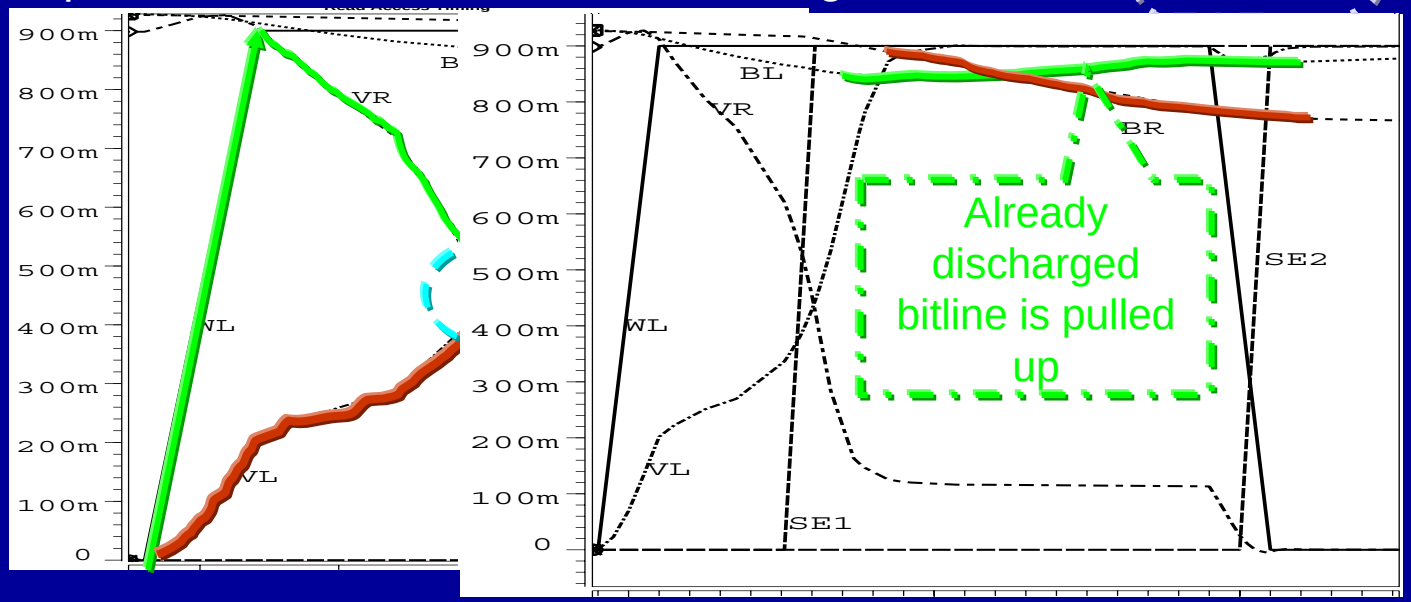
March test sequence comparison

Logic Fault Models	Conventional Test Sequences			Proposed Sequences	
	March C-	March B	March SR	Opt. March C-	March Q
Address Decoder Fault	+	+	-	+	-
Data Retention Fault	-	-	+	+	+
Low Supply Data Retention Fault	-	-	-	+	+
Stuck-at Fault	+	+	+	+	+
Transition Fault	+	+	+	+	+
Random Read Fault	+-	+-	+-	+-	+-
Read Destructive Fault	+	+	+	+	+
Deceptive Read Destructive Fault	-	-	+	+	+
Incorrect Read Fault	+	+	+	+	+
State Coupling Fault	+	-	+	+	+
Disturb Coupling Fault	+	-	+	+	+
Incorrect Read Coupling Fault	+	-	+	+	+
Read Destructive Coupling Fault	+	-	+	+	+
Transition Coupling Fault	+	+-	+	+	+-
Test Time	10N	17N	14N	12N	10N

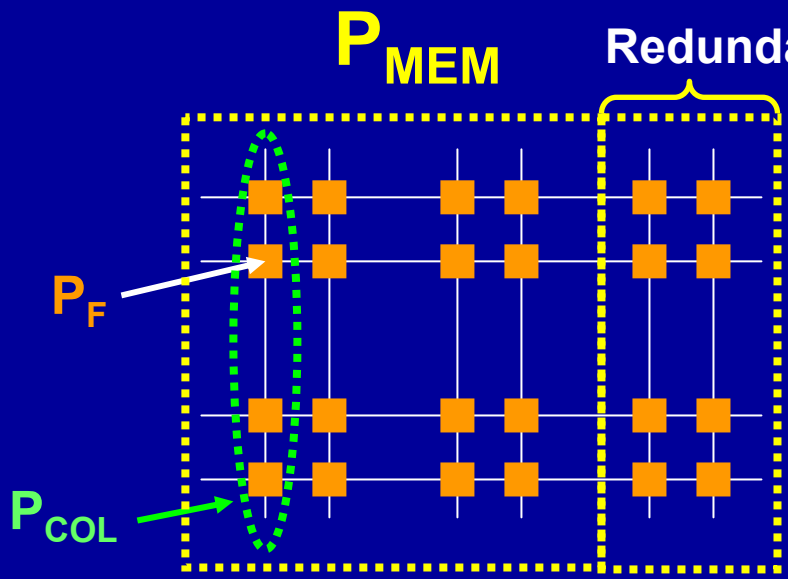
Double Sensing: A Novel DFT Circuit to Reduce Test Time

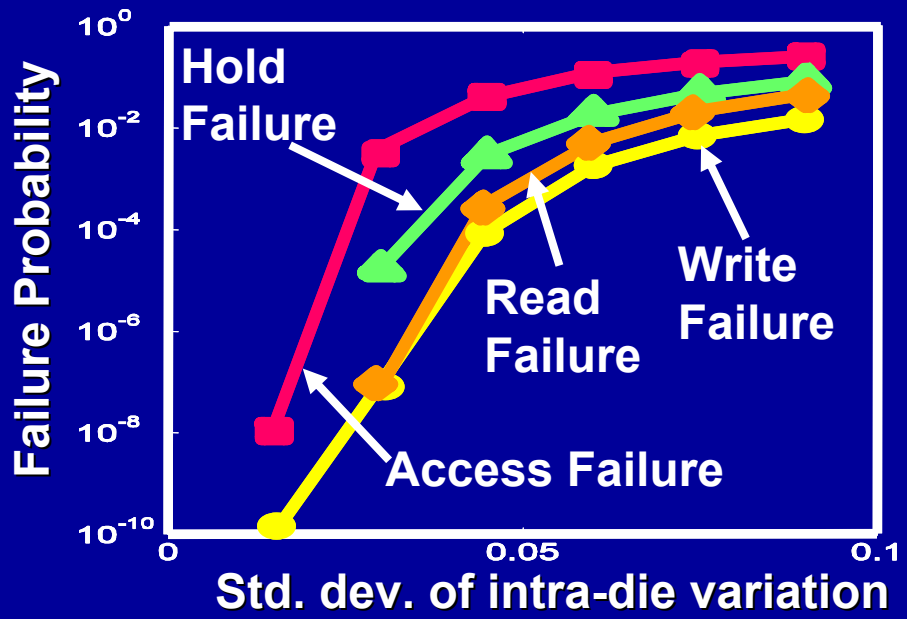
- In order to reduce test time, double sensing is used to detect Deceptive Read Destructive Fault in one cycle
  - However, the WL can NOT be extended too long for test since the cell may flip after the WL is activated.
  - Another parallel sense amp is fired at the time the WL is activated to obtain the bitline differential voltage.
  - Then the required WL extension time is minimized.

The flipping is sensed .



Parametric Failures and Yield





Inter-die Variation

SRAM die with a global shift in process parameter

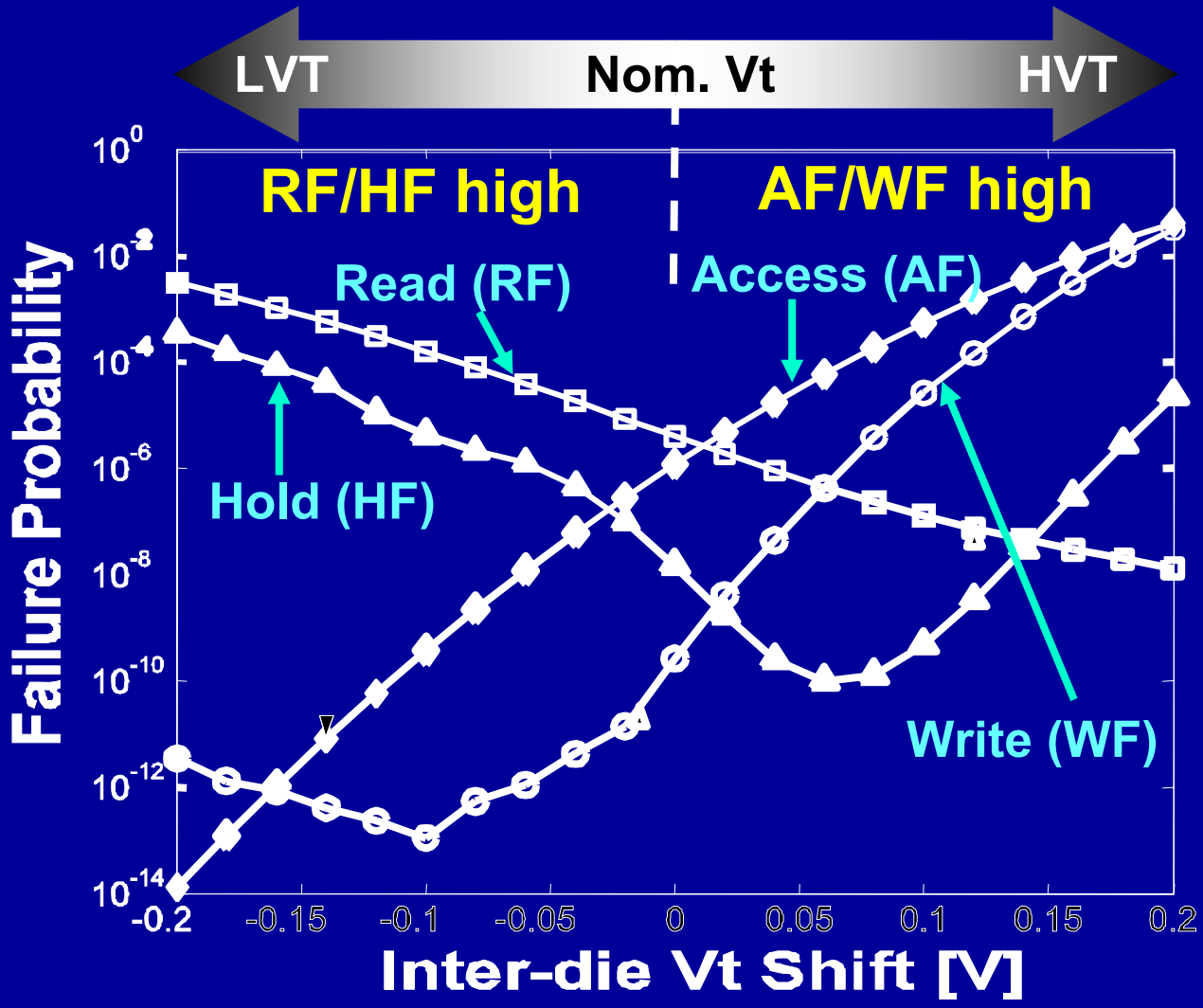
Intra-die Variation => Cell, Column and Memory failure

$$P_{MEM} = f(\textit{Inter-die Variation})$$

$$\textit{Yield} = \frac{\#\ \textit{Non-Faulty Die}}{\#\ \textit{Total Die}}$$

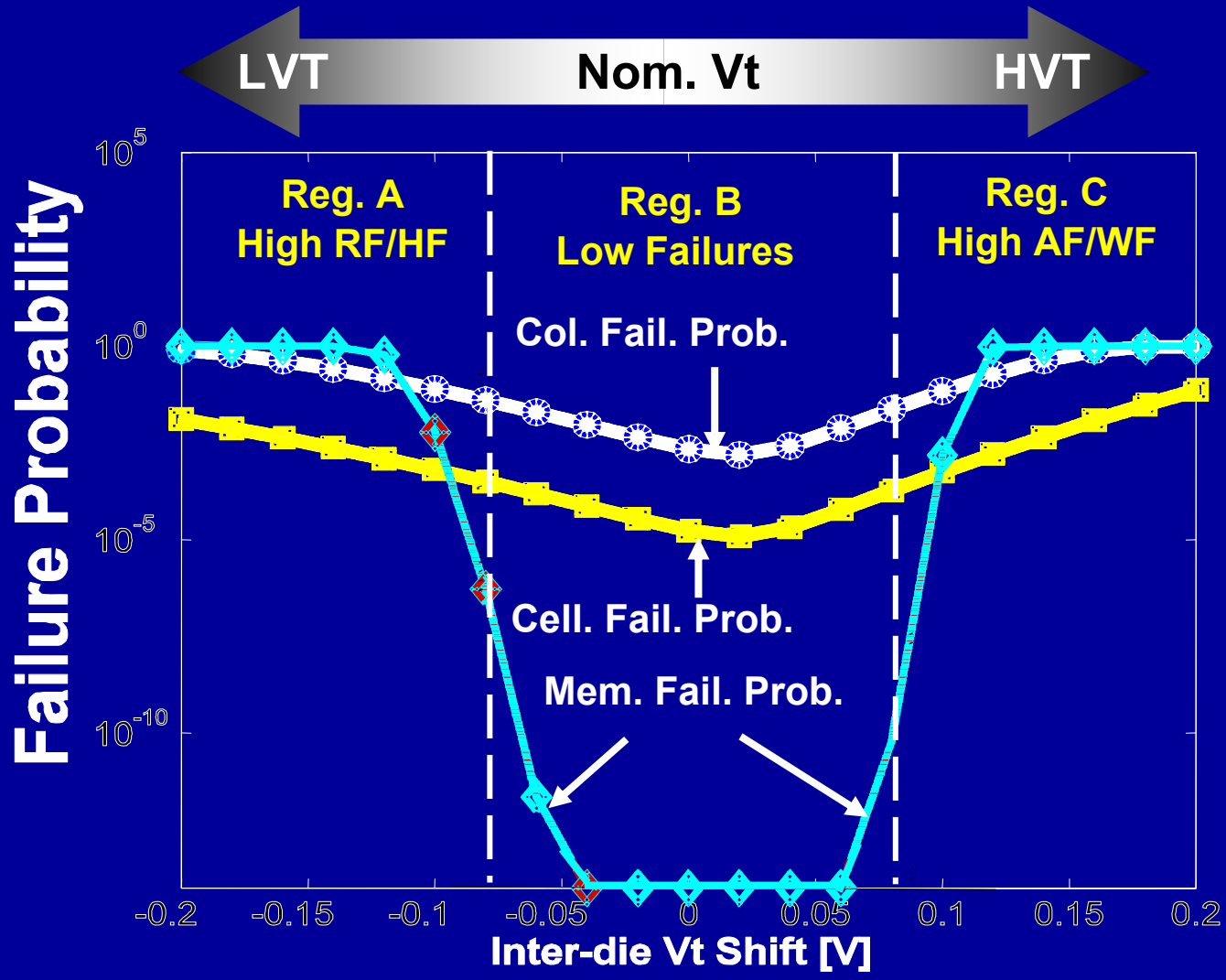
What is the effect of Inter-die variation on Parametric Failures?

Inter-die Variation and Cell Failure



- Inter-die shift in process parameter amplifies the failure due to intra-die variations.

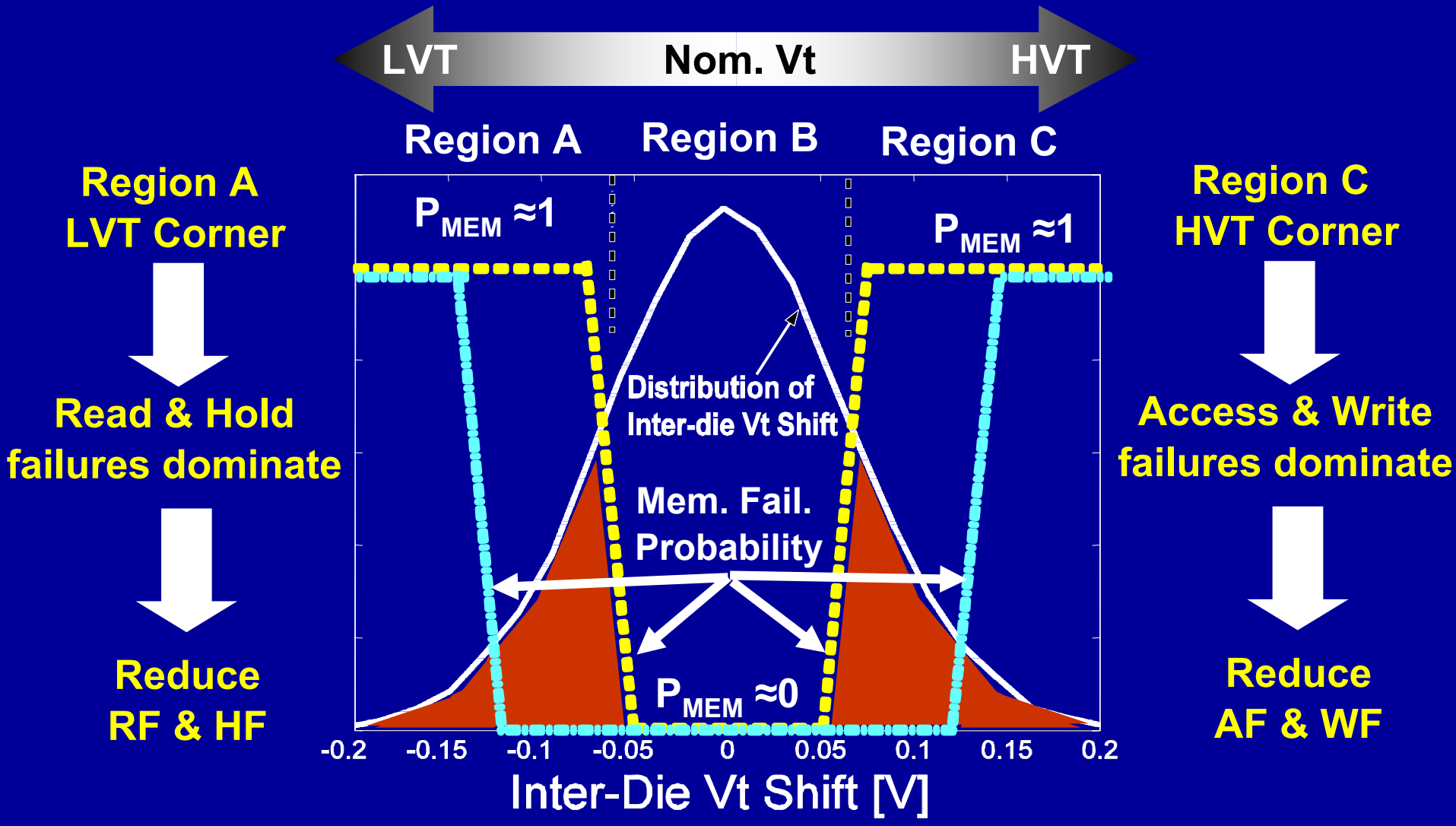
Inter-die Variation and Memory Failure



- Memory failure probabilities are high at high when inter-die (global) shift in process is high.

How can we improve yield considering both inter-die and intra-die variations?

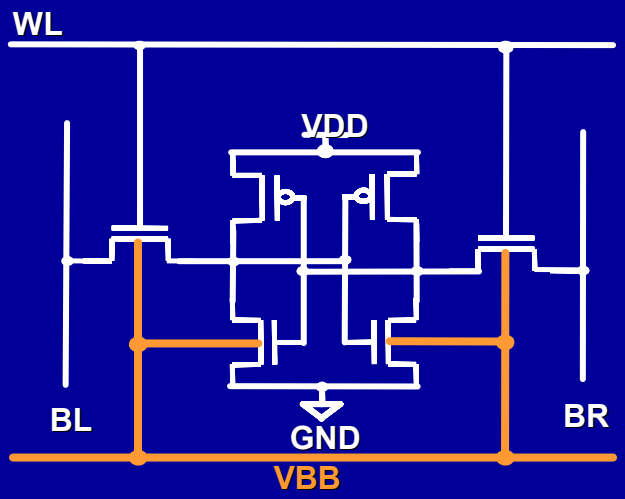
Adaptive Repairing of SRAM Array

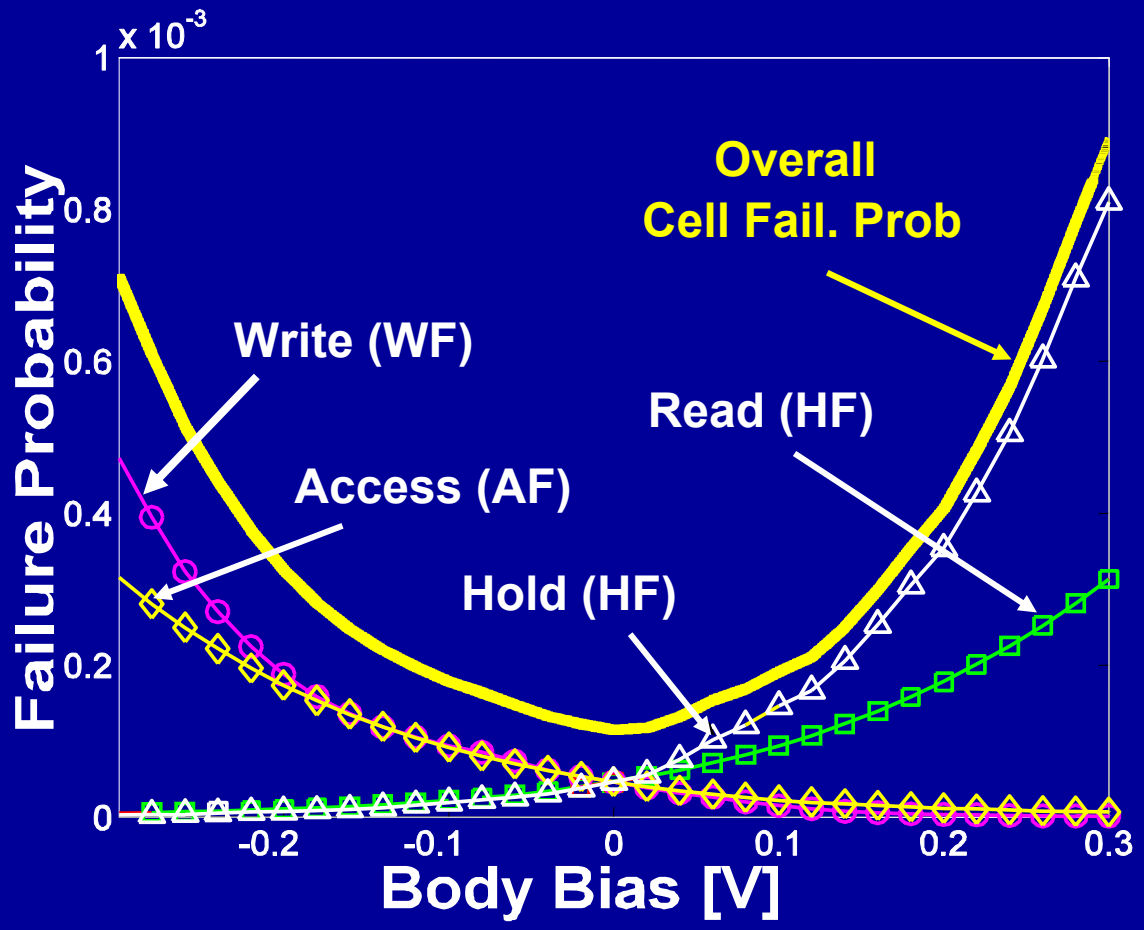


- Reduce the dominant failures at different inter-die corners to increase width of low failure region.

How can we reduce the dominant failures at different inter-die corners?

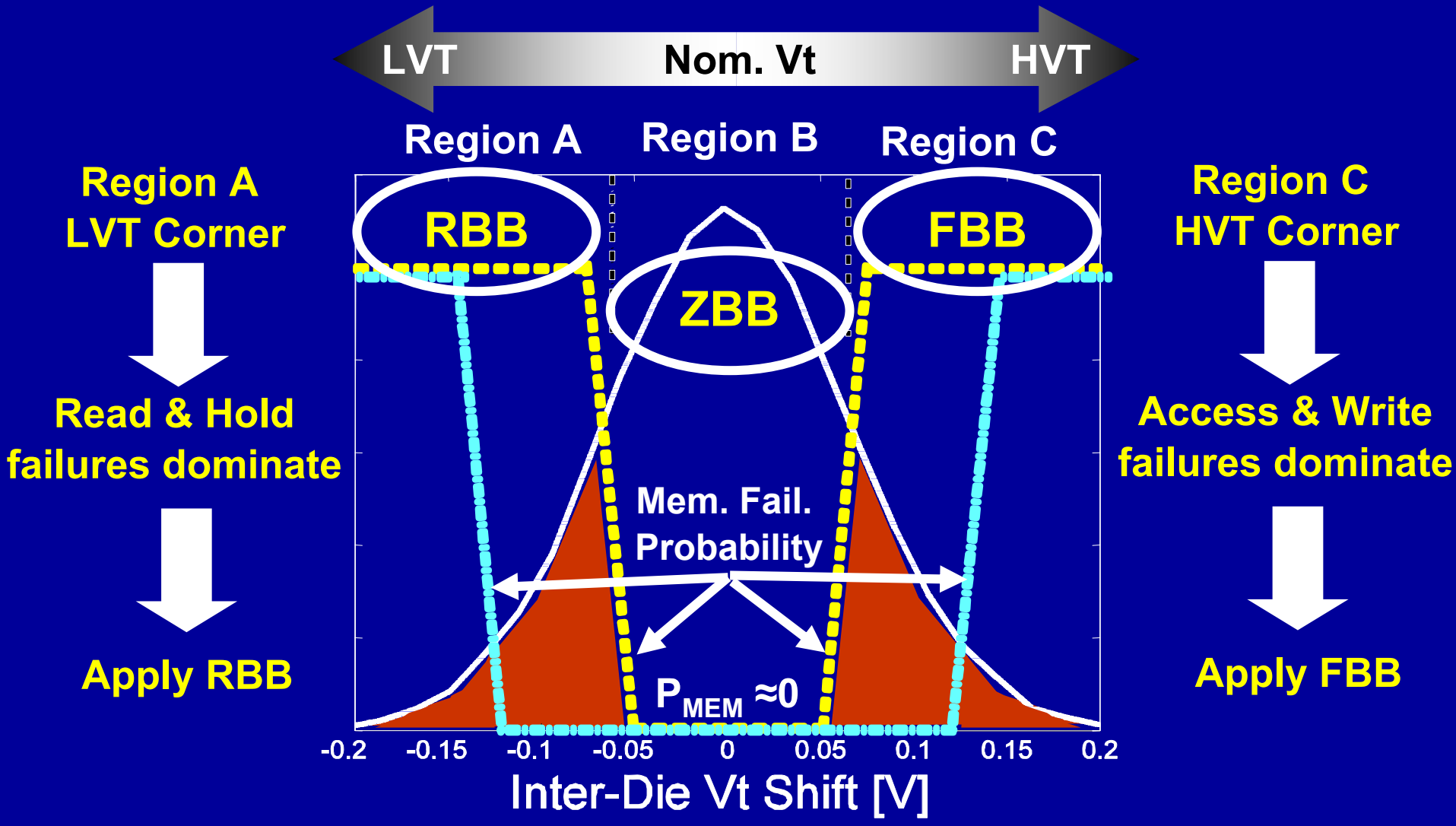
Body Bias and Parametric Failures





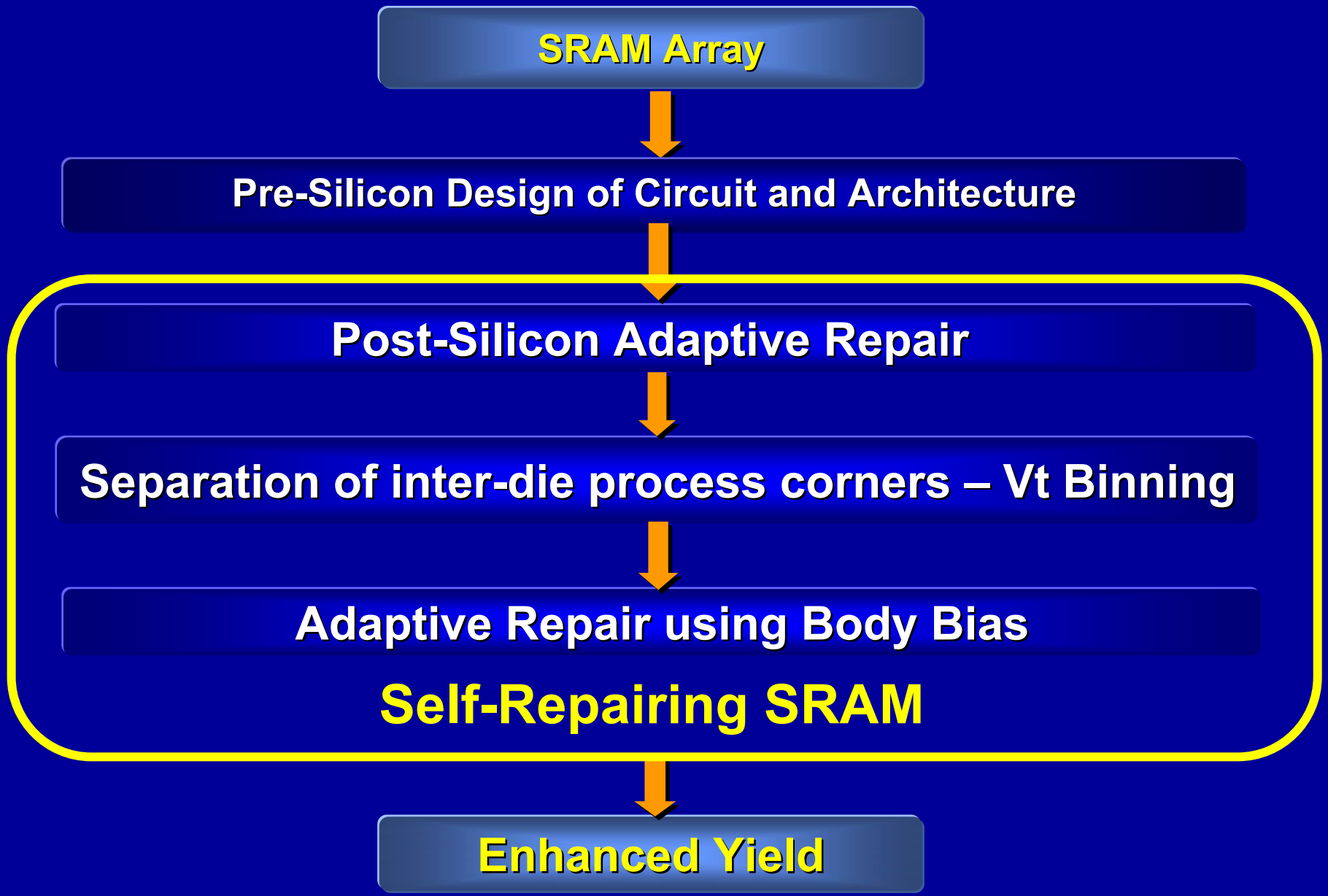
- Proper body bias can reduce parametric failures
  - –Forward bias reduces Access & Write failures
  - –Reverse bias reduces Read & Hold failures

Adaptive Repair using Body Bias



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

Self-Repair Technique in SRAM



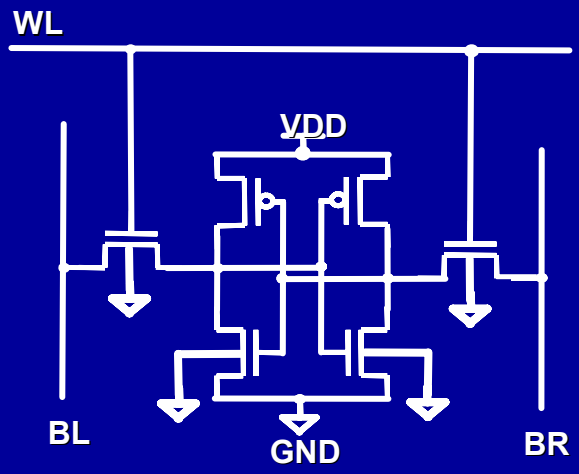
How we can identify the inter-die Vt corner under a large random intra-die variation ?

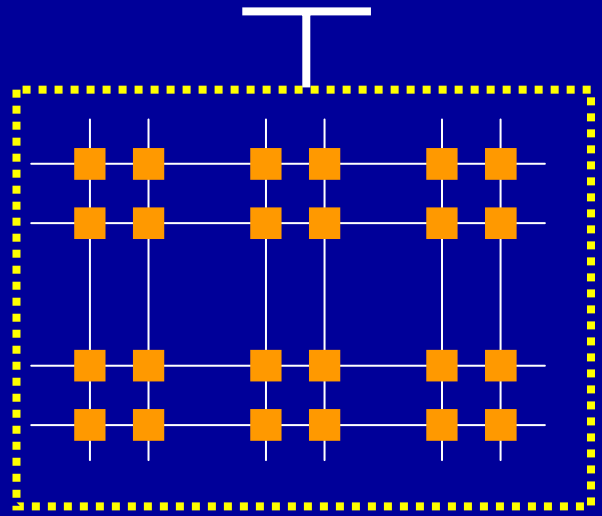
- Monitor circuit parameters e.g. delay and leakage
  - –Effect of inter-die variation can be masked by that of intra-die variation
- Adding a large number of random variables reduces the effect of intra-die variation

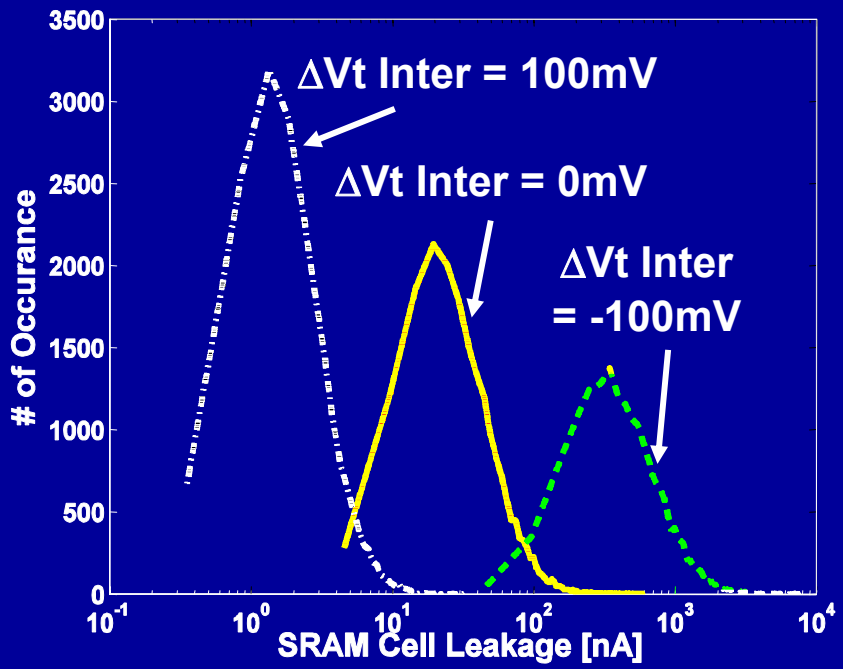
$$Y = \sum_{i=1}^{n} X_i$$

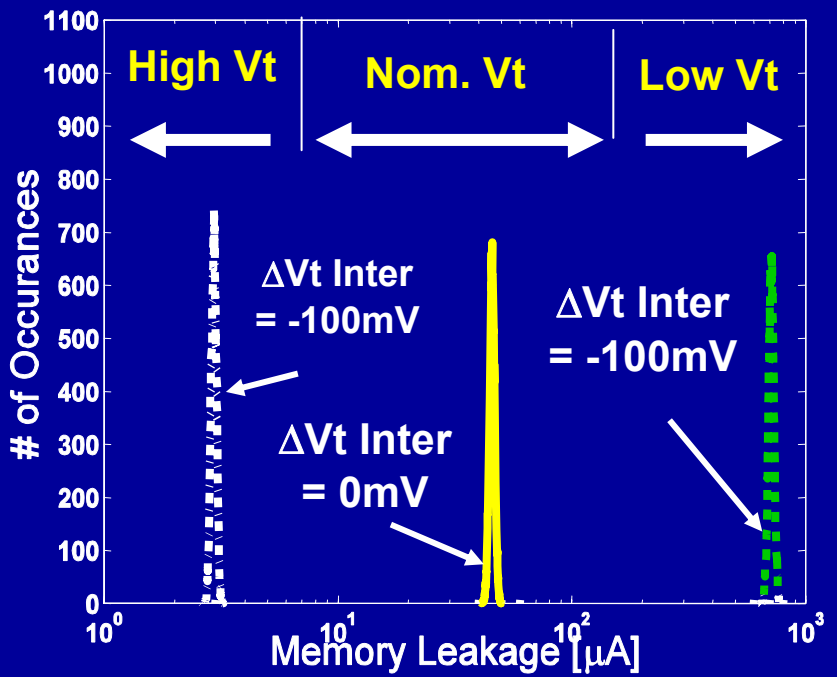
$$\Rightarrow \mu_Y = \sum_{i=1}^{n} \mu_{Xi} = N\mu_X \;\&\; \sigma_Y^2 = \sum_{i=1}^{n} \sigma_{Xi}^2 = N\sigma_X^2 \Rightarrow \frac{\sigma_Y}{\mu_Y} = \frac{1}{\sqrt{N}} \frac{\sigma_X}{\mu_X}$$

Vt Binning by Leakage Monitoring

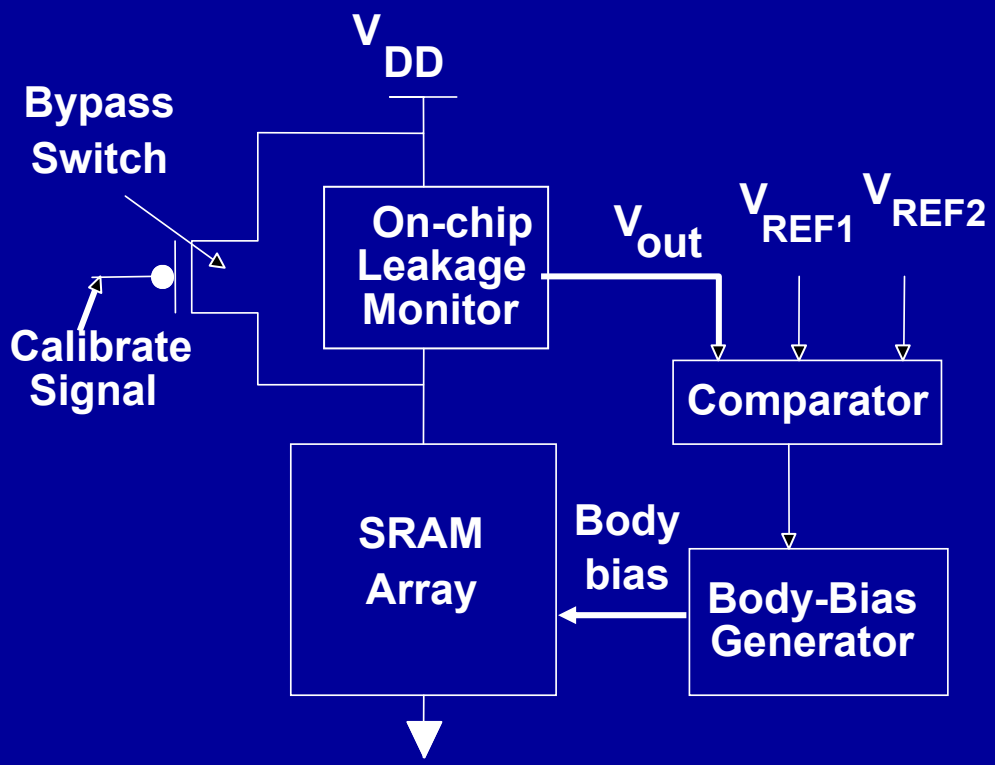


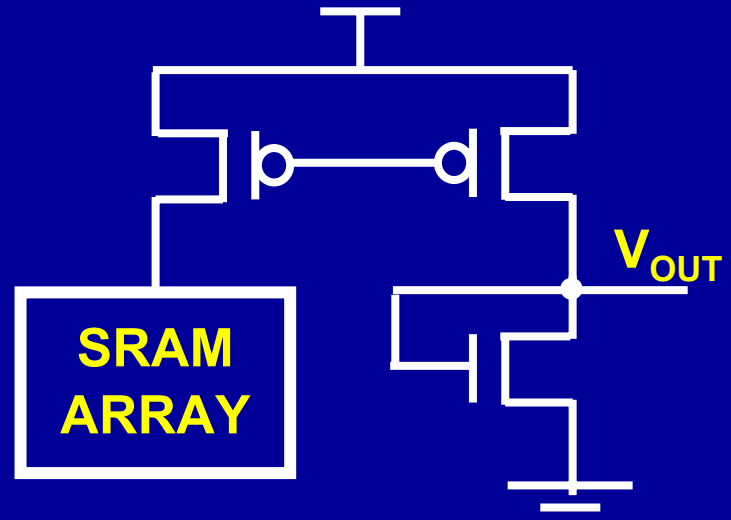


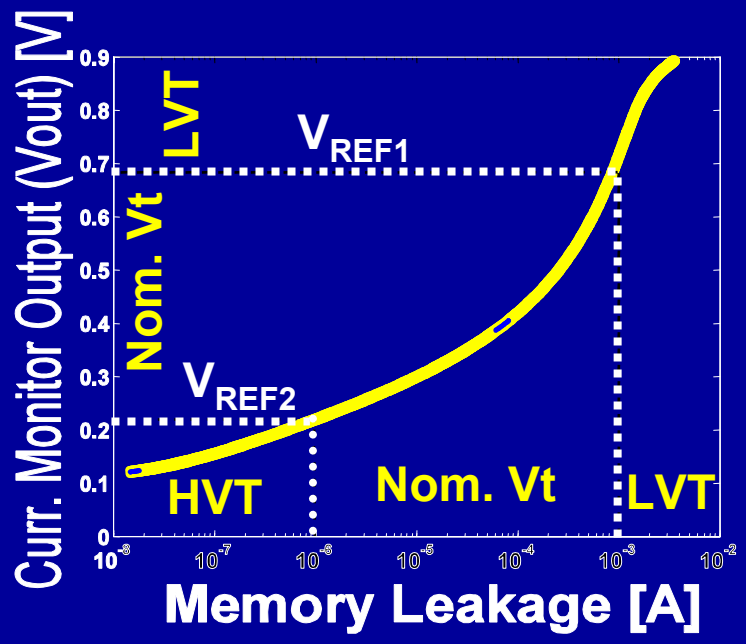




Self-Repair using Leakage Monitoring



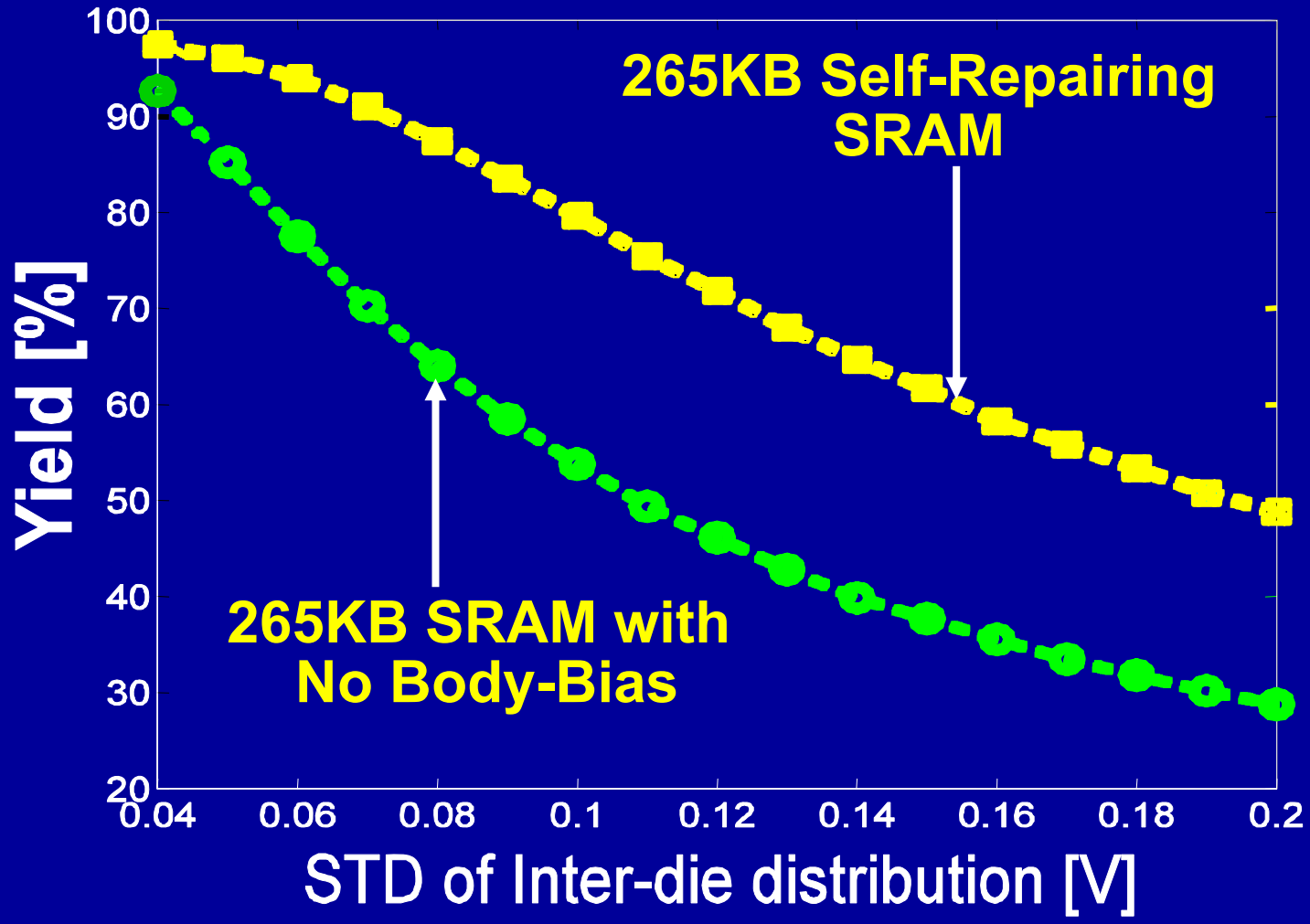




- On-chip monitoring of leakage of entire array
- Body-bias is generated based on leakage monitor output
- Leakage monitored is bypassed in normal operating mode

Current Monitor Circuit

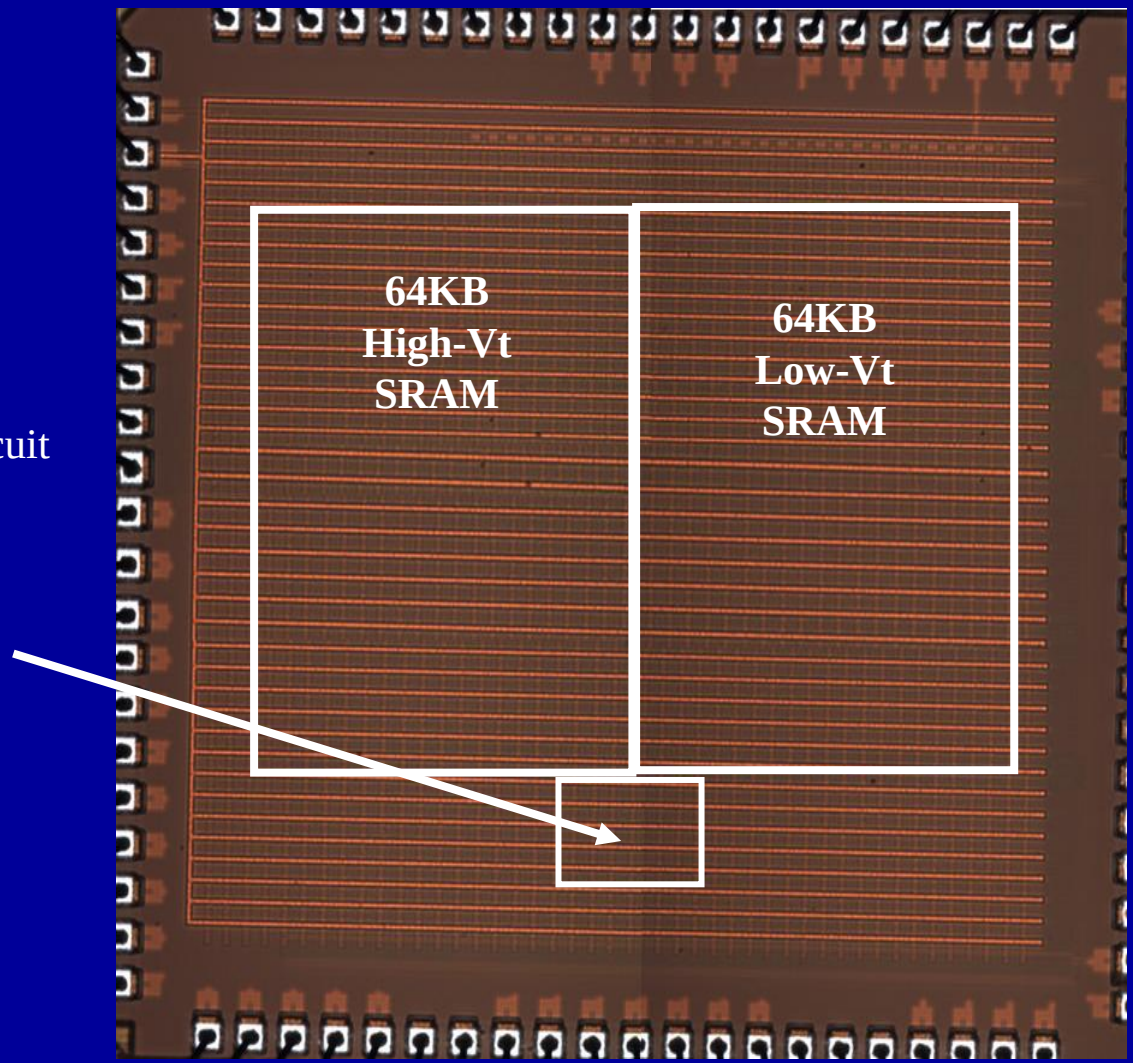
Yield Enhancement using Self-Repair



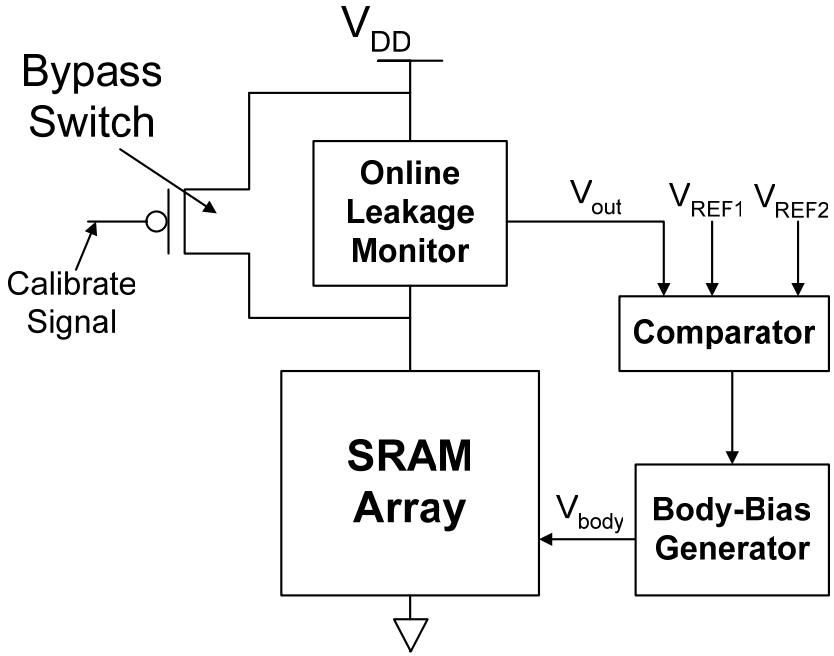
- **Self-Repairing SRAM using body-bias can significantly improve design yield.**

Self-Repairing SRAM: Die Photo

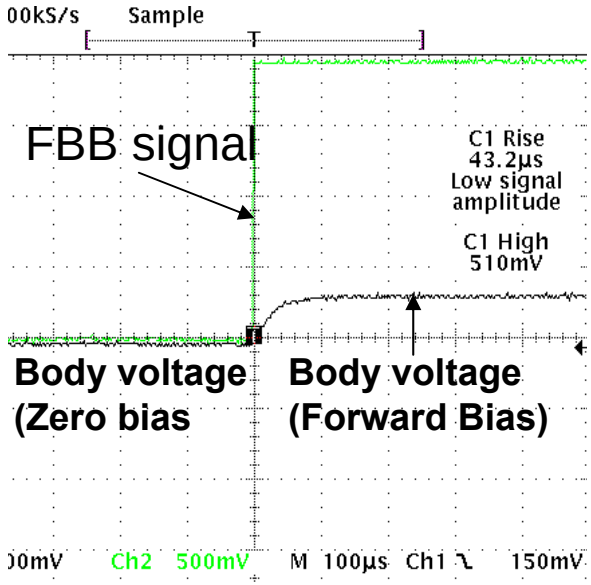
Self-Repair Circuit



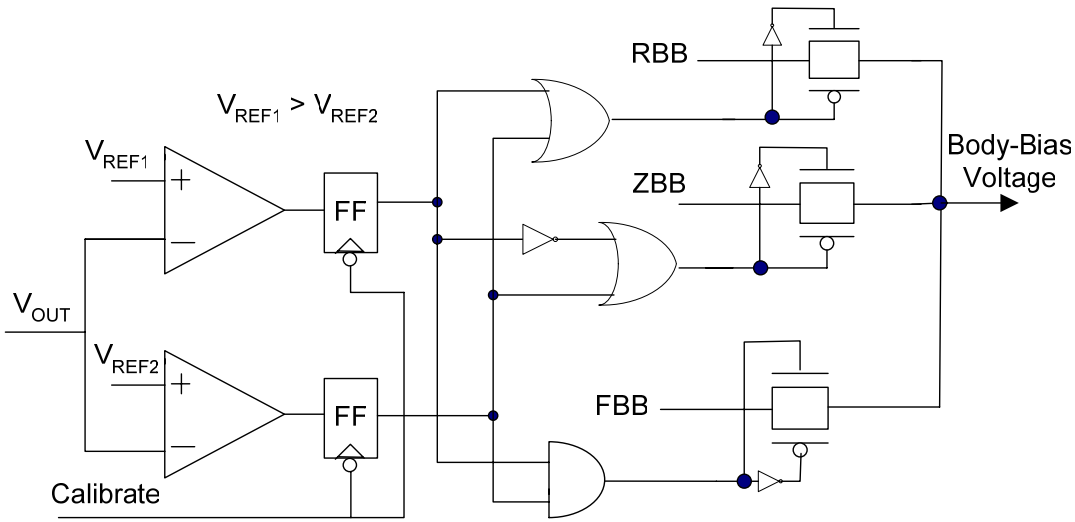
Schematic and measurement of self-repair mechanisms



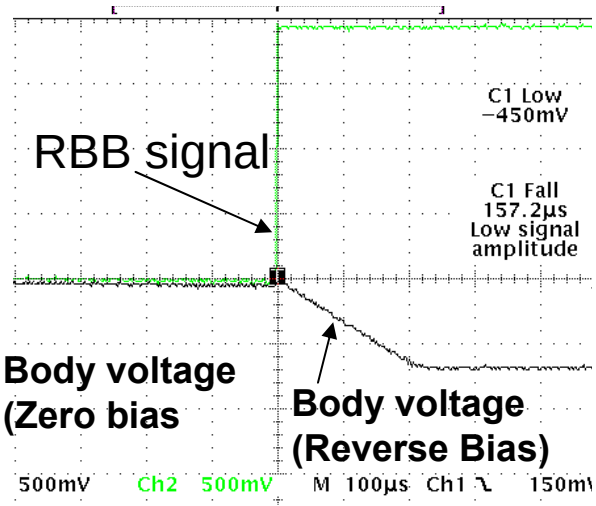
Schematic of the Self-repairing SRAM



Measured waveform of body voltage for Forward bias with On-chip FBB generator

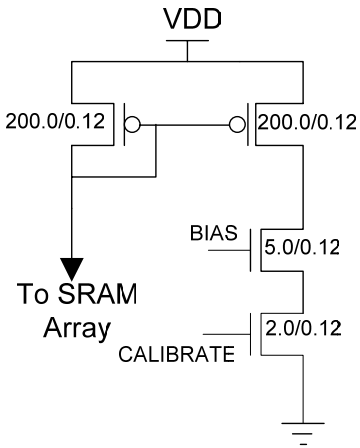


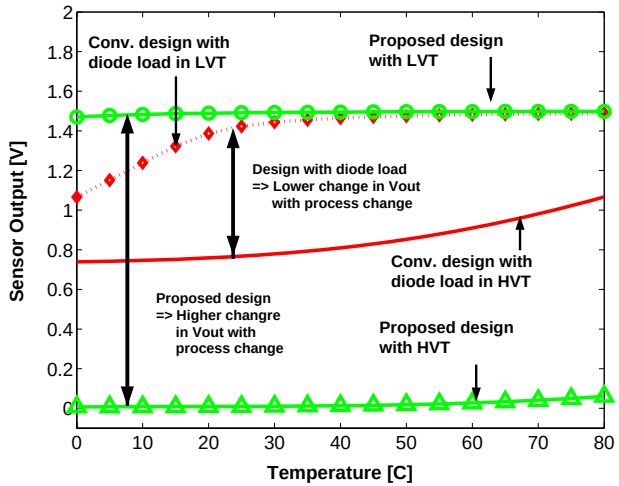
Bodybias generation logic (MUX switches are designed with level converter for negative bias, not shown here)



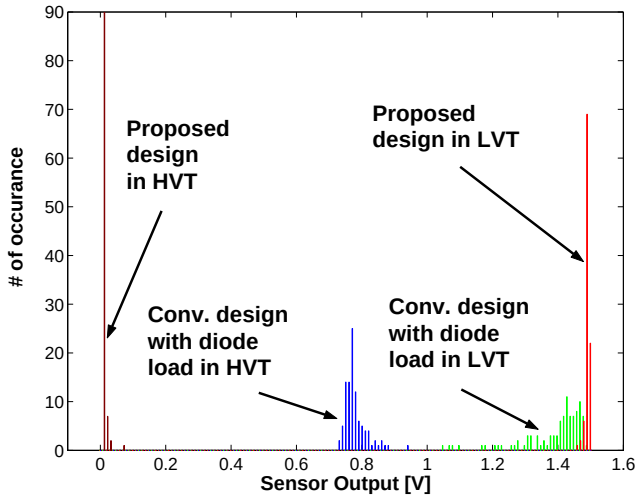
Measured waveform of body voltage for Reverse bias with On-chip RBB generator

Design and Measurement of Current Sensor

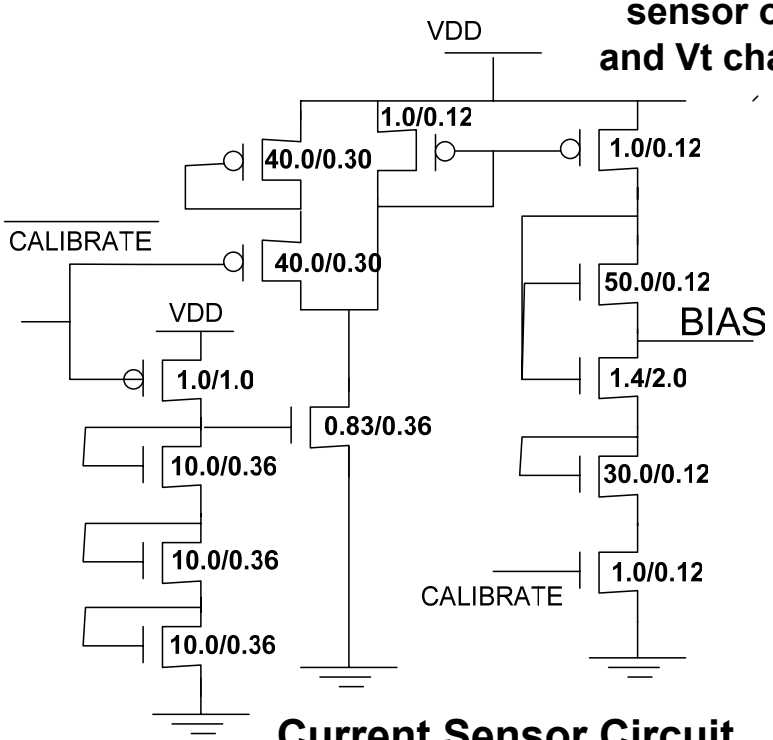




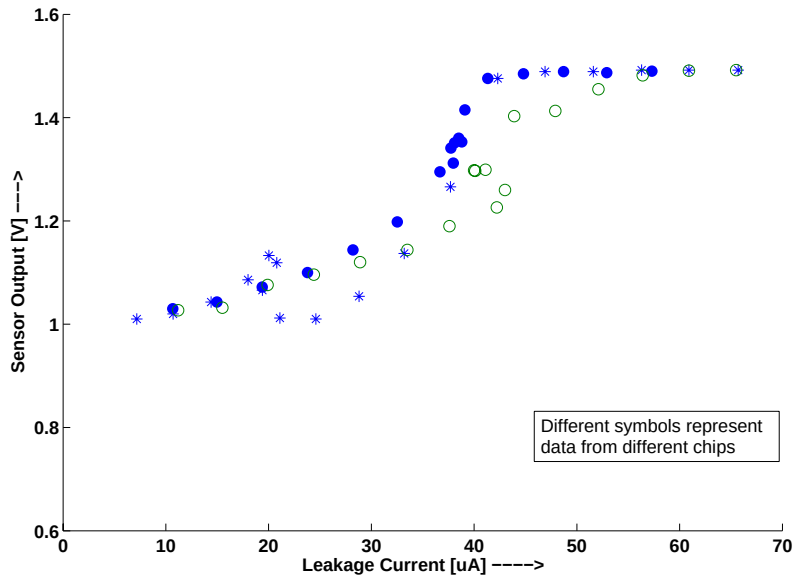
sensor output with temperature and Vt change (simulation 0.13μm)



Effect of intra-die variation on sensor output (simulation 0.13μm)

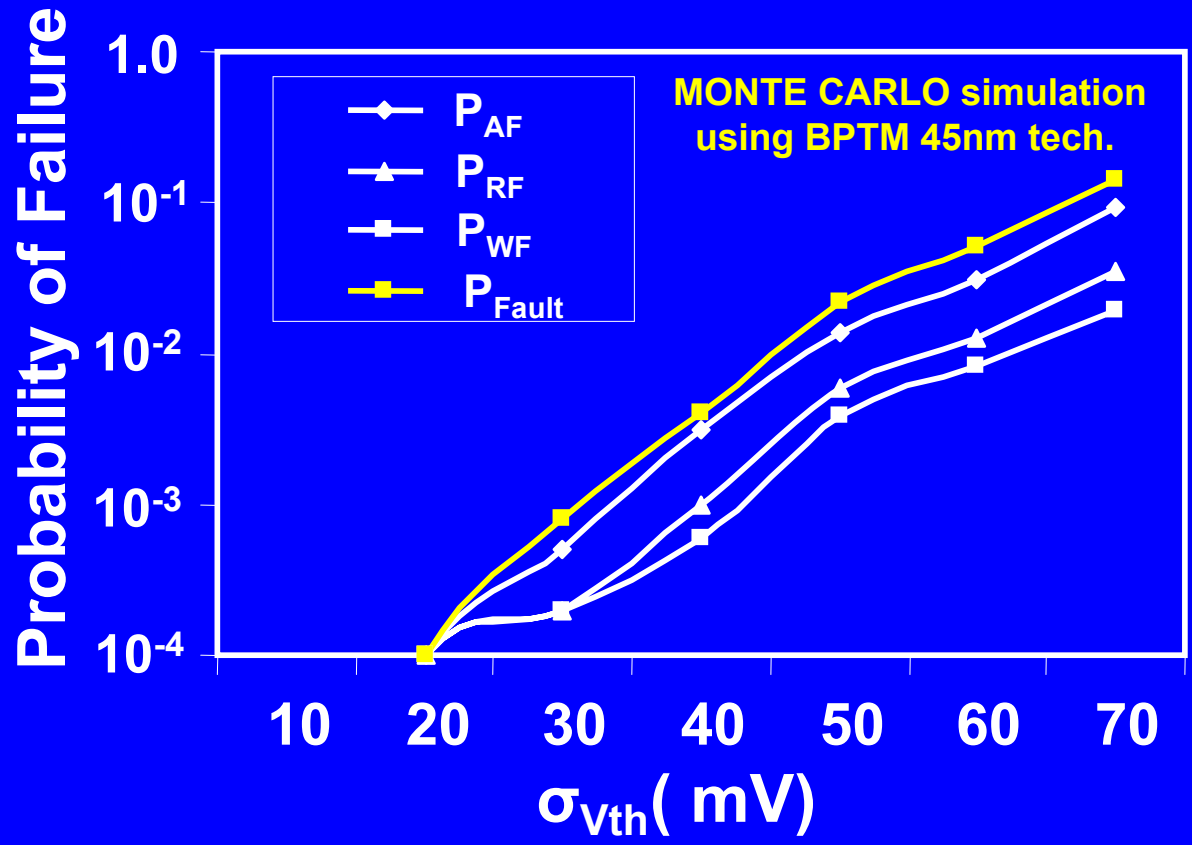


Current Sensor Circuit



Measured current sensor output attached to the 64KB LVT array

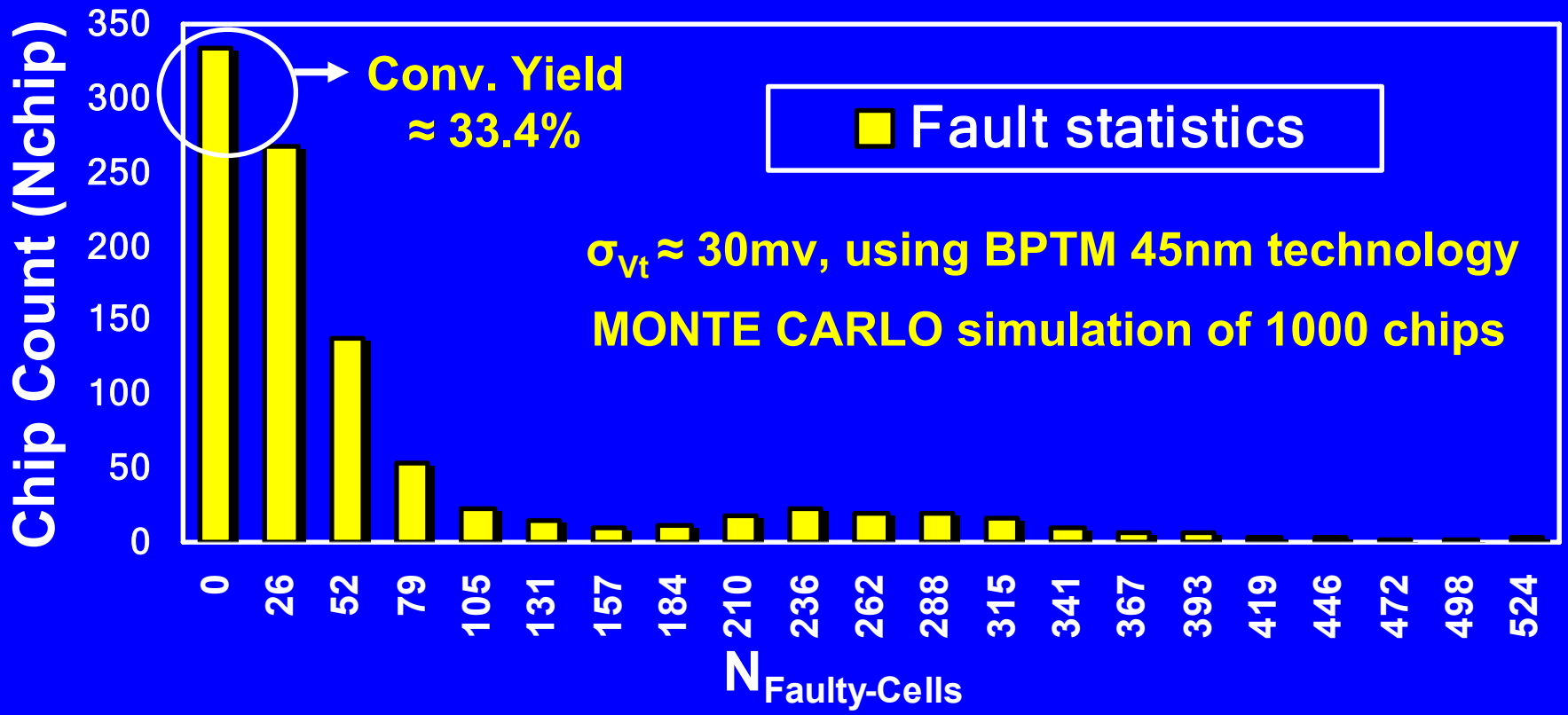
Failure Probability in SRAM Memory Cells



- Intrinsic Fluctuation of V_{th} due to random dopant effect
- $P_{Fault} = P_{AF}\ U\ P_{RF}\ U\ P_{WF}$
- In 45nm technology $\sigma_{Vth} \approx$ 30mV $\rightarrow$ $P_{Fault} > 1.0 \times 10^{-3}$

Large number of faulty cells in nano-scale SRAM under process variation

Fault Statistics in 32K Cache

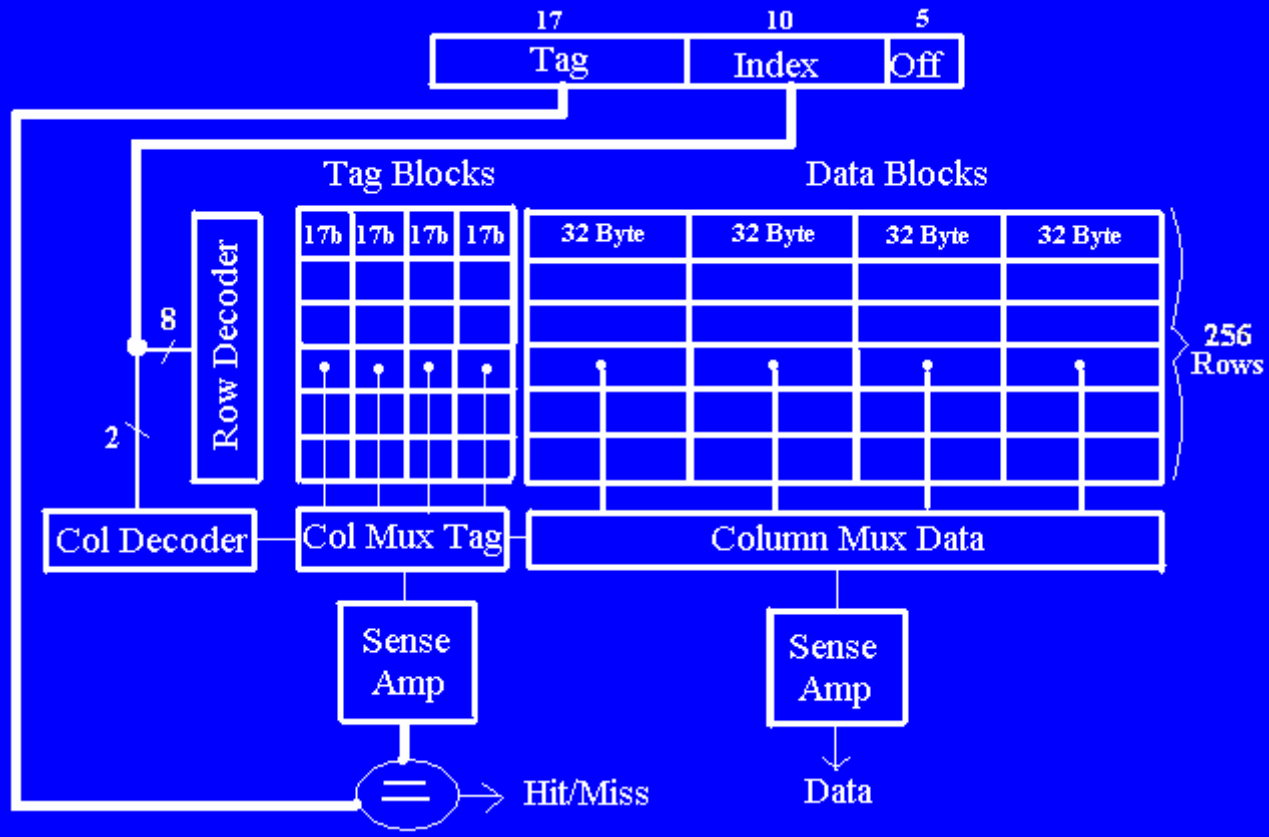


$N_{Faulty\text{-}Cells} = P_{Fault} \; X \; N_{Cells}$ (total number of cells in a cache)

- Conventional 32K cache results in only 33.4% yield

Need a process/fault-tolerant mechanisms to improve the yield in memory

Basic Cache Architecture



- Index = Row Address + Column Address
- Multiple cache blocks are stored in a single row
  - Minimize delay, area, routing complexity
- Column MUX selects one block

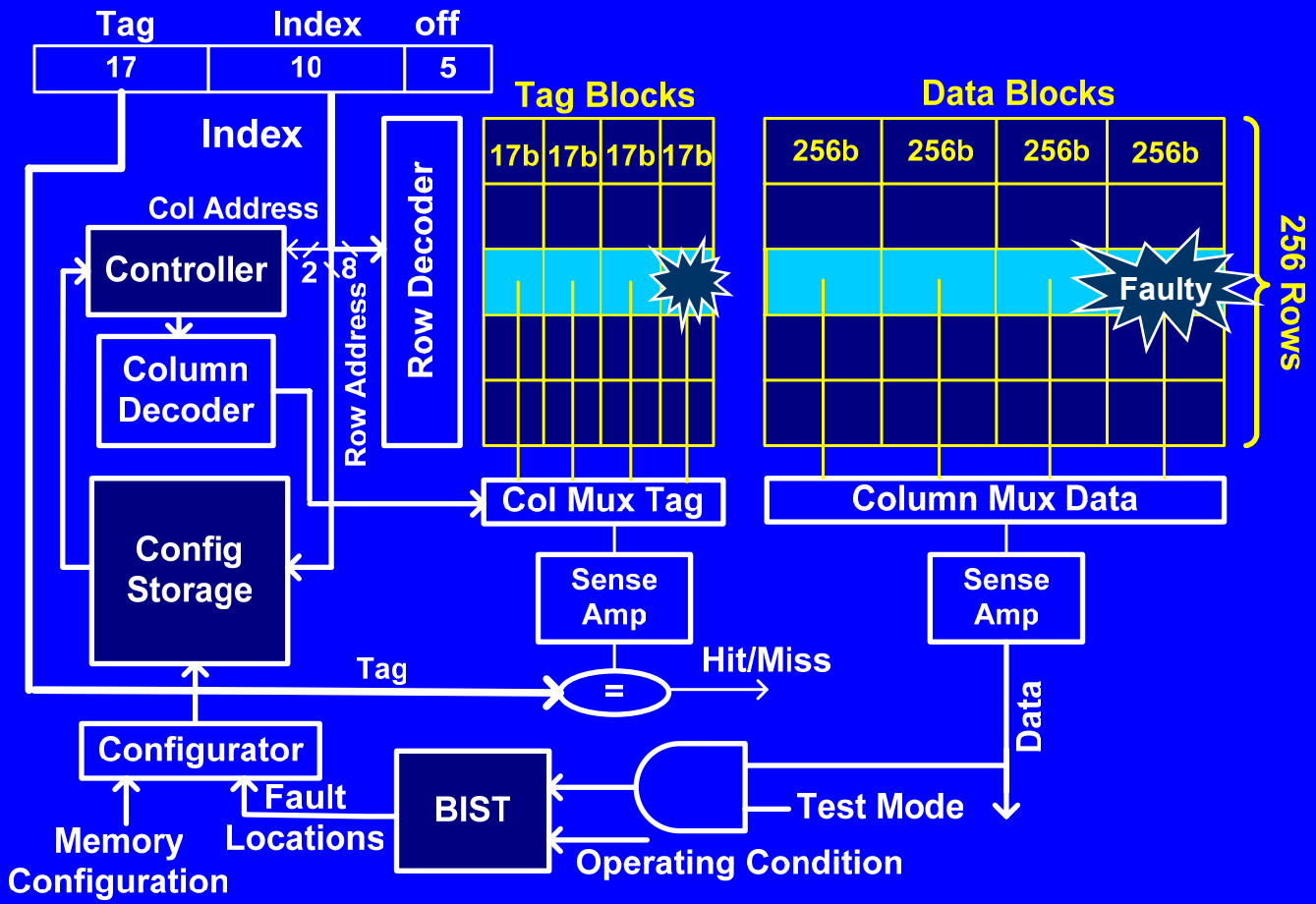
Basic Cache Architecture

BPTM 45nm technology, 32KByte direct mapped cache

# of Block in a Row	1 Block	2 Blocks	4 Blocks
Decoder Delay	0.086ns	0.085ns	0.084
Wordline Delay	0.069ns	0.075ns	0.128ns
Bitline to Q Delay	0.452ns	0.355ns	0.313ns
Total Delay	0.608ns	0.515ns	0.525ns
Energy	0.166nJ	0.181nJ	0.195nJ

- For 32K cache best # of cache blocks/row = 2
- We choose 4 blocks in a row for our design
  - Results in higher yield – 16.25% increase
  - 2% cache access penalty
  - 7% energy overhead

Fault-Tolerant Cache Architecture



- BIST detects the faulty blocks
- Config Storage stores the fault information

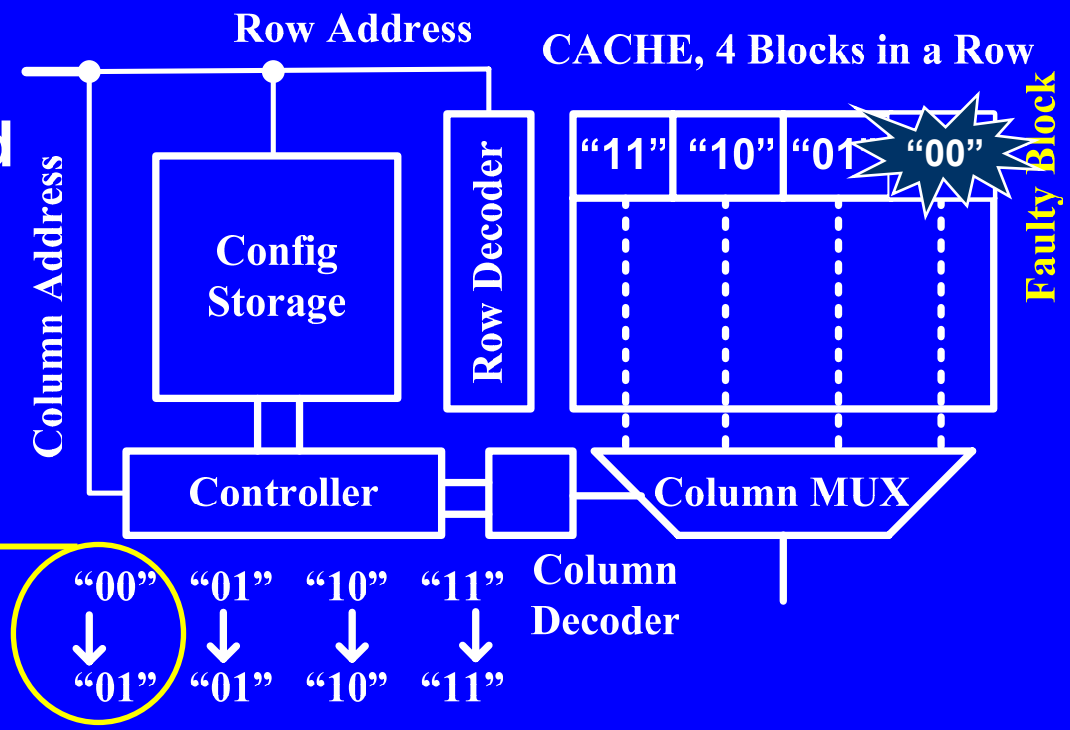
Idea is to resize the cache to avoid faulty blocks during regular operation

Resizing the Cache

Config Storage is accessed in parallel with cache

Feeds the fault information to controller

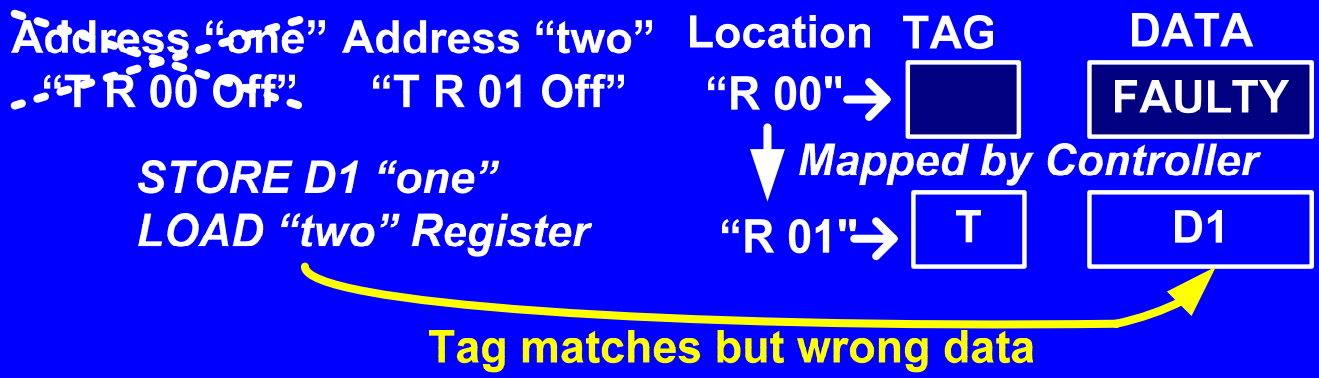
Controller alters the column address



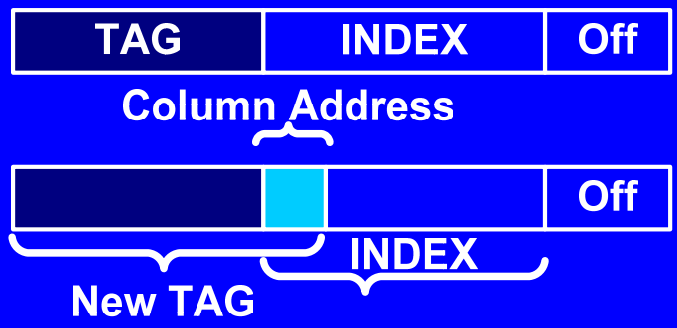
Force the column MUX to select a non-faulty block in the same row if the accessed block is faulty

Handle large number of faults without significantly reducing the cache size

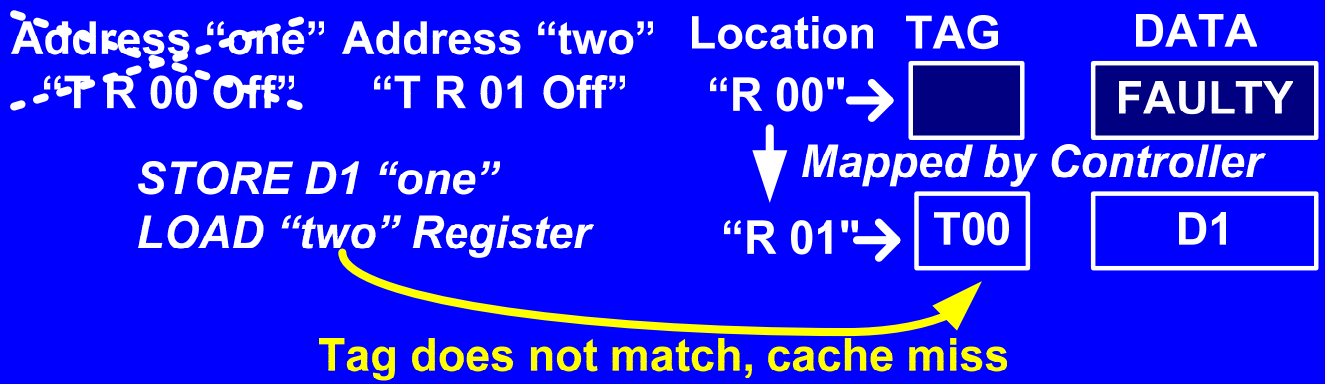
Mapping Issue



More than one INDEX are mapped to same block

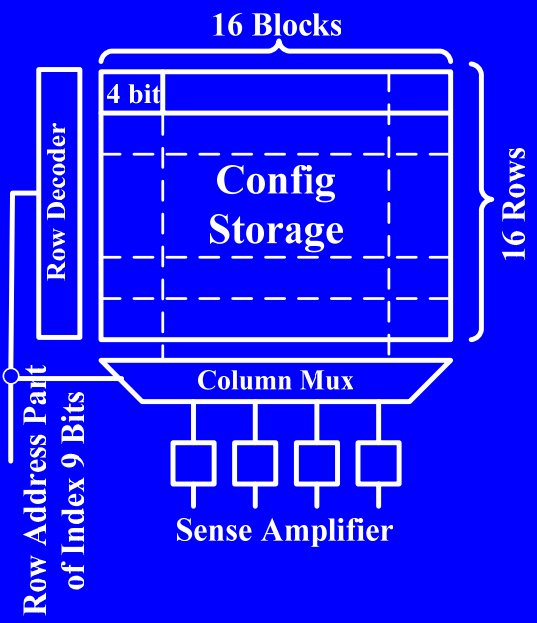


Include column address bits into TAG bits



Resizing is transparent to processor → same memory address

Config Storage



	Cache 32KByte	Config Storage 1Kbit
Blocks per row	4	x
Block Size	32Byte	4bit

Accessed in parallel with cache

4 bit fault information about 4 blocks stored in a single cache row

- One bit fault information per cache block
- Bits are determined by BIST at the time of testing
- Accessed using row address part of INDEX
- Provides the fault information of all the blocks in a cache row to controller

Controller

Column address selection based on fault location

Faulty Blocks in Accessed Row	Fault Information by Config Storage	Accessed Column Address			
		00	01	10	11
		Forced Column Address			
		↓	↓	↓	↓
None	0000	00	01	10	11
3rd Block	0010	00	01	00	11
2nd &3rd Block	0110	00	00	11	11
1st, 2nd & 3rdBlock	1110	11	11	11	11
All four Blocks	1111	NA	NA	NA	NA

Based on 4 bits read from Config Storage controller alters the column address

Controller

Column address selection based on fault location

Faulty Blocks in Accessed Row	Fault Information by Config Storage	Accessed Column Address			
		00	01	10	11
		Forced Column Address			
		↓	↓	↓	↓
None	0000	00	01	10	11
3rd Block	0010	00	01	00	11
2nd &3rd Block	0110	00	00	11	11
1st, 2nd & 3rdBlock	1110	11	11	11	11
All four Blocks	1111	NA	NA	NA	NA

One block in a row is faulty

Selects the first available non-faulty block
e.g 3rd block → 1st block

Controller

Column address selection based on fault location

Faulty Blocks in Accessed Row	Fault Information by Config Storage	Accessed Column Address			
		00	01	10	11
		Forced Column Address			
		↓	↓	↓	↓
None	0000	00	01	10	11
3rd Block	0010	00	01	00	11
2nd &3rd Block	0110	00	00	11	11
1st, 2nd & 3rdBlock	1110	11	11	11	11
All four Blocks	1111	NA	NA	NA	NA

Two blocks in a row is faulty

Selects two non-faulty blocks respectively

e.g 2nd block → 1st block

3rd block → 4th block

Controller

Column address selection based on fault location

Faulty Blocks in Accessed Row	Fault Information by Config Storage	Accessed Column Address			
		00	01	10	11
		Forced Column Address			
		↓	↓	↓	↓
None	0000	00	01	10	11
3rd Block	0010	00	01	00	11
2nd &3rd Block	0110	00	00	11	11
1st, 2nd & 3rdBlock	1110	11	11	11	11
All four Blocks	1111	NA	NA	NA	NA

Three blocks in a row is faulty

All the blocks are mapped to non-faulty block, e.g 4th block

One non-faulty block in each row, this architecture can correct any number of faults

Energy, Performance, and Area Overhead of Config Storage and Controller

BPTM 45nm technology, 32KByte Cache, 1Kbit Config Storage

Energy and Performance	32KB Cache	Config Storage & Controller
Delay (ns)	0.45	0.22
Area overhead	NA	0.5%
Energy overhead	NA	1.8%

- Controller changes the column address before data reaches at column MUX
- Does not affect the cache access time
- Negligible energy and area overhead (excluding BIST)

Results: P_{op} (ECC, Redundancy and Proposed Scheme)

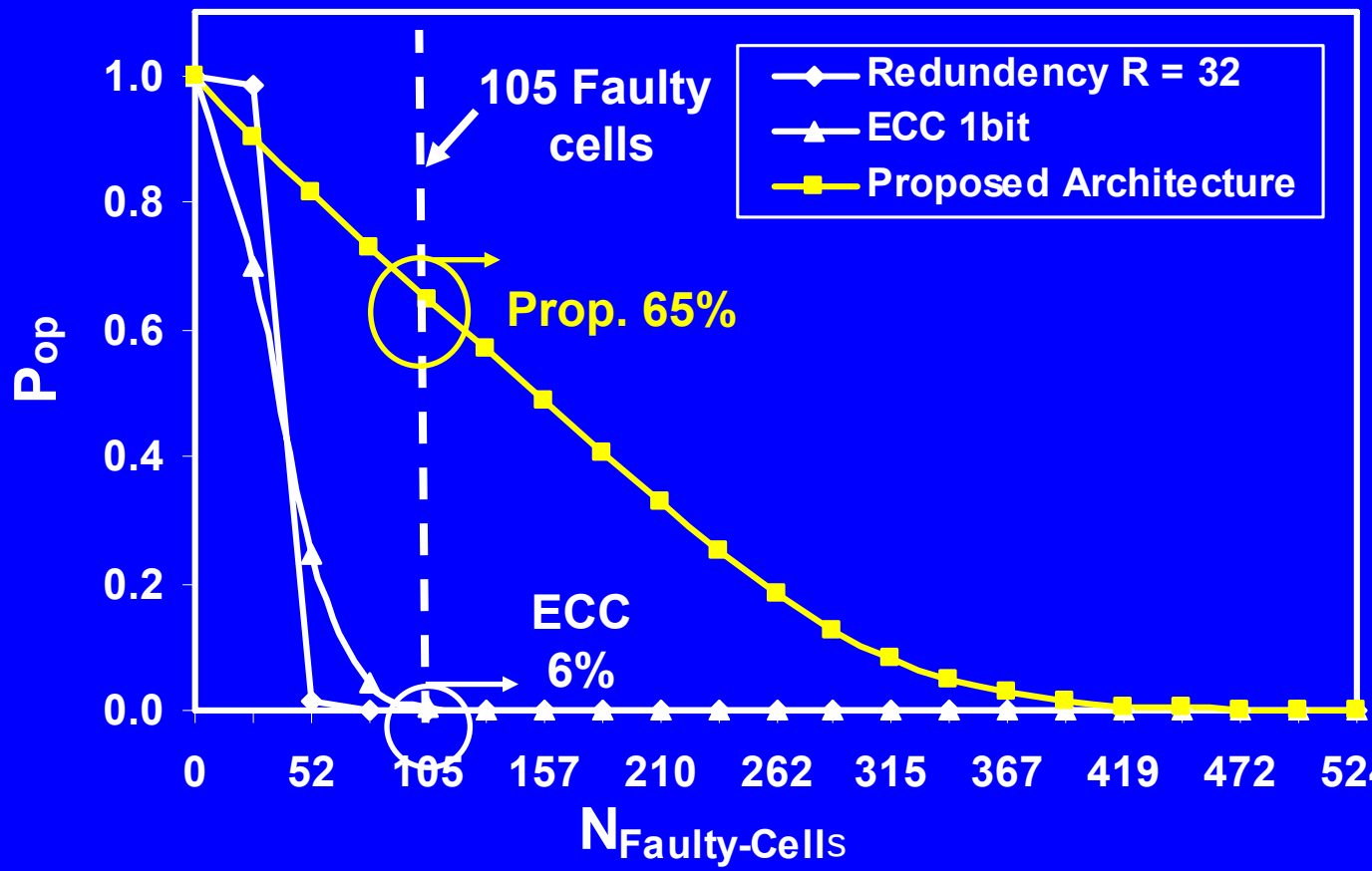
- P_{op}: Probability that a chip with $N_{Faulty\text{-}cells}$ can be made operational
- Faults are randomly distributed across chip
- Yield is defined as:

$$Y_{chip} = \frac{1}{N_{Tot}} \sum_{N_{Faulty\_Cell}} P_{op}(N_{Faulty\_Cell}) * N_{chip}(N_{Faulty\_Cell})$$

- Each scheme add some extra storage space
  - P_{op} includes the probability of having faults in these blocks
  - To consider area, yield is redefined as:

$$Y_{chip}^{eff} = Y_{chip} \frac{A_{chip\_without\_any\_scheme}}{A_{chip\_with\_fault\_tolerant\_scheme}}$$

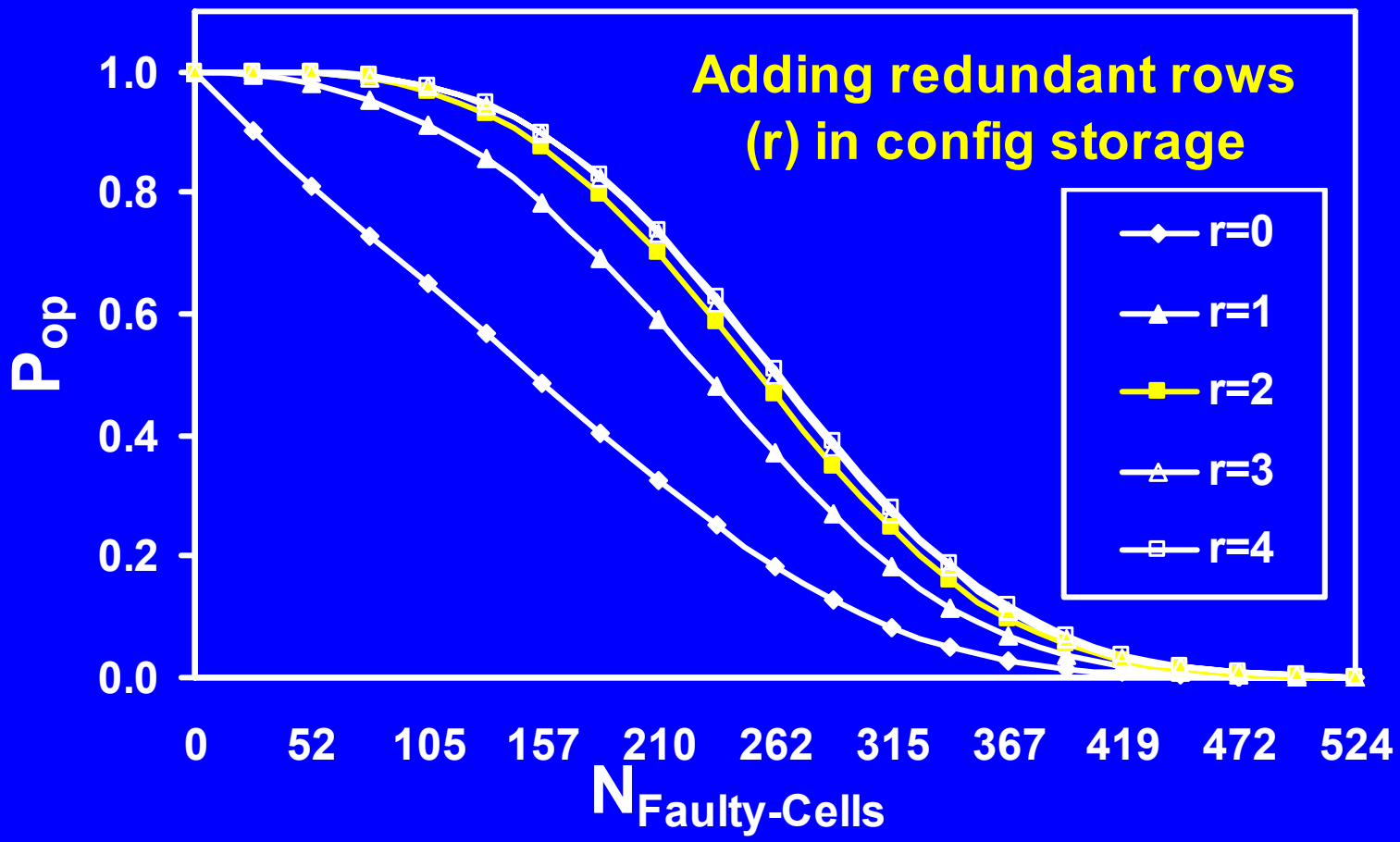
Results: P_{op}



% of the chips with 105 faulty cells which can be saved by

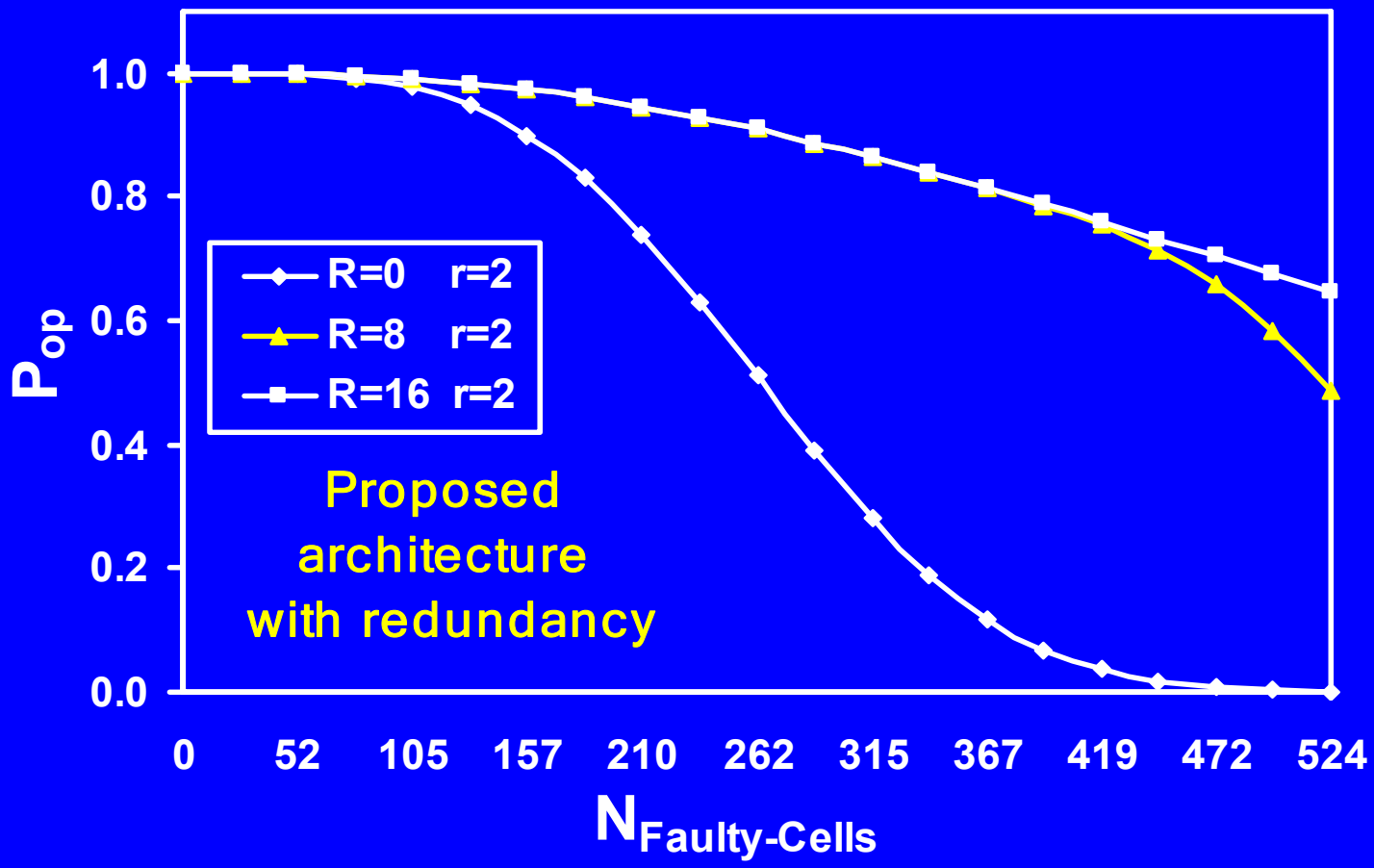
- **Proposed scheme ~ 65% (high fault tolerant capability)**
- **ECC ~ 6%**
- **Redundancy ~ 0%**

Results: P_{op}



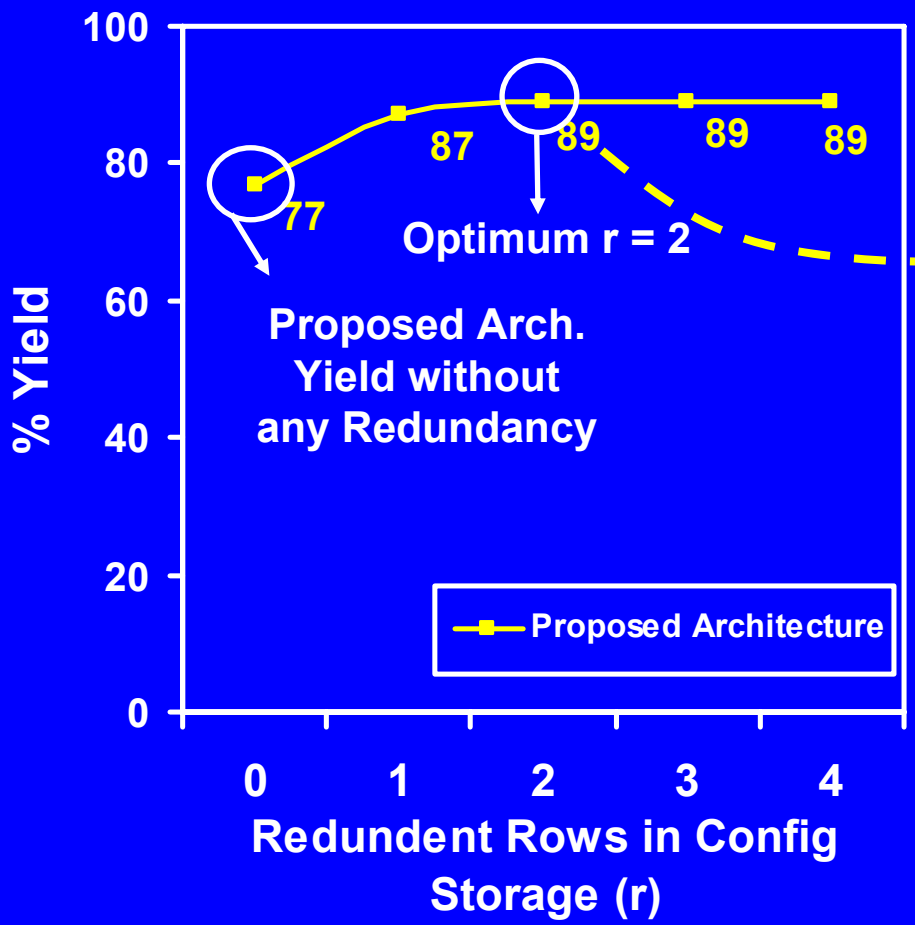
- P_{op} improves with redundant rows in config storage
- r = 2 is optimum for 32K cache with 1Kbit config storage

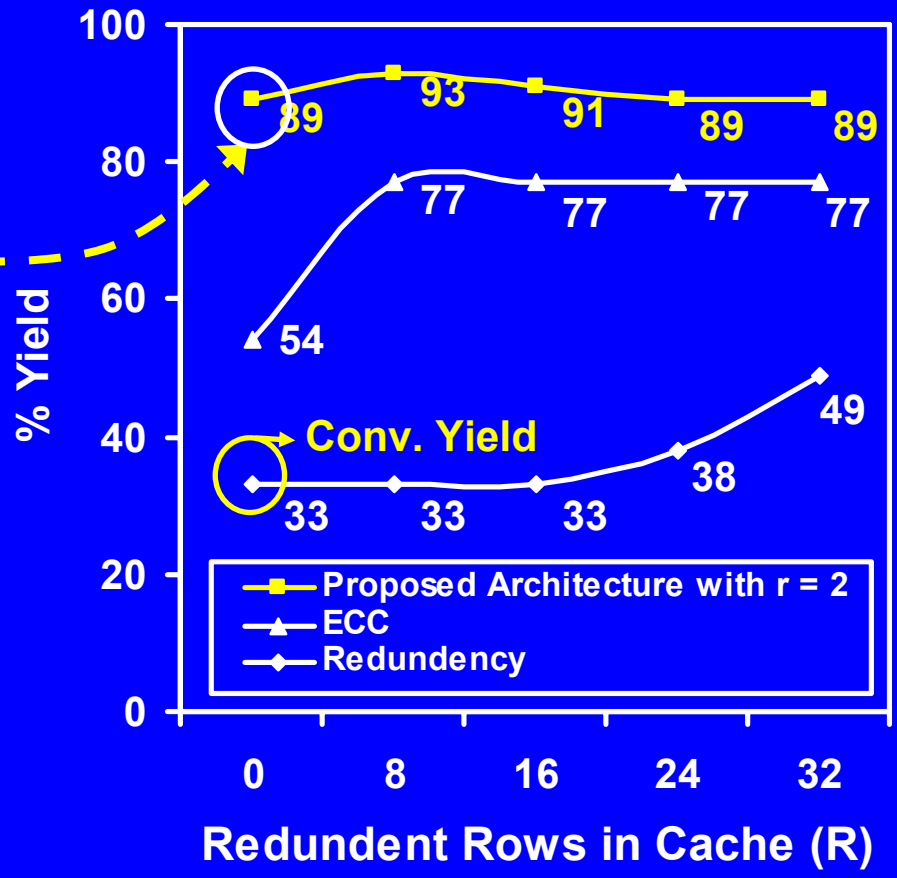
Results: P_{op}



- Adding redundant rows (R) in cache in proposed scheme improves the P_{op} further
  (optimum is R =8 for 32K cache)

Effective Yield of 32K Cache

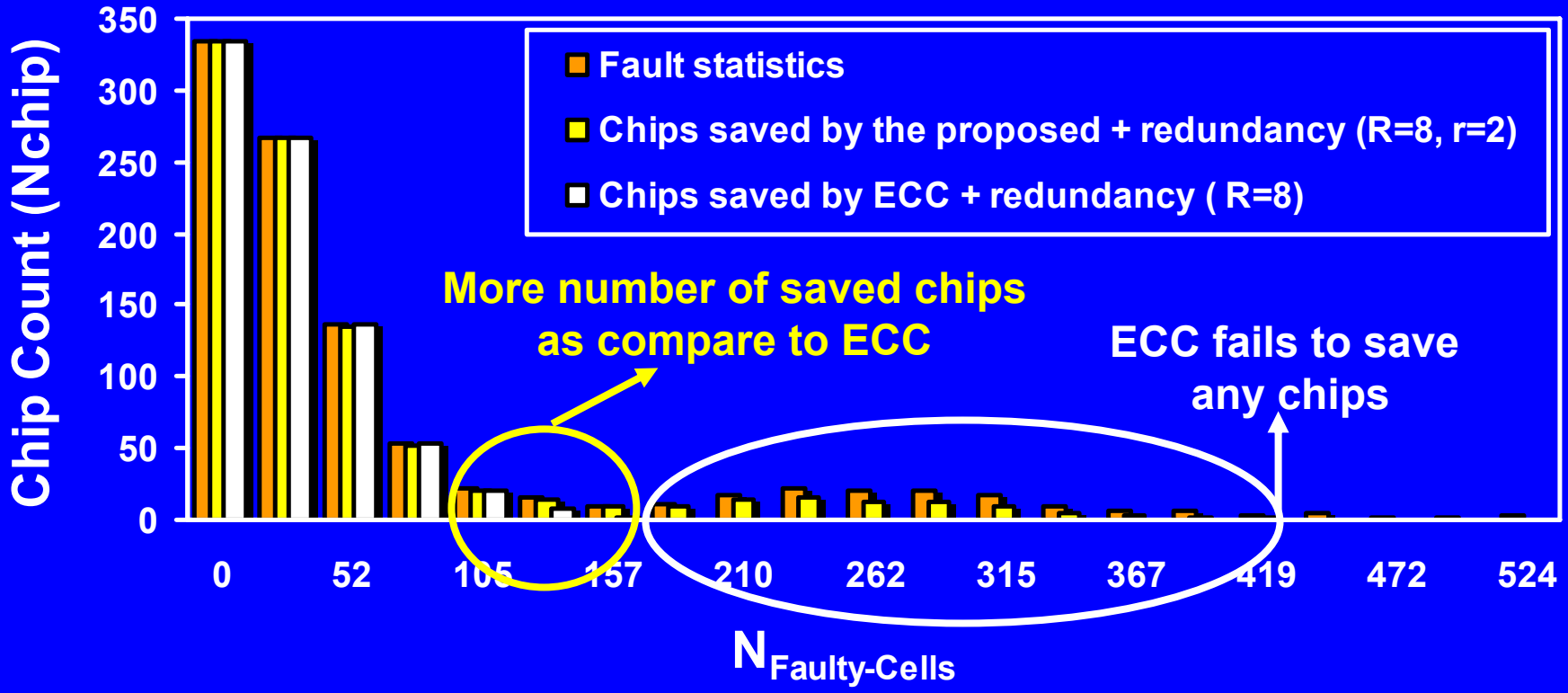




- ECC + Redundancy yield ~ 77%
- Proposed architecture + Redundancy yield ~ 93%

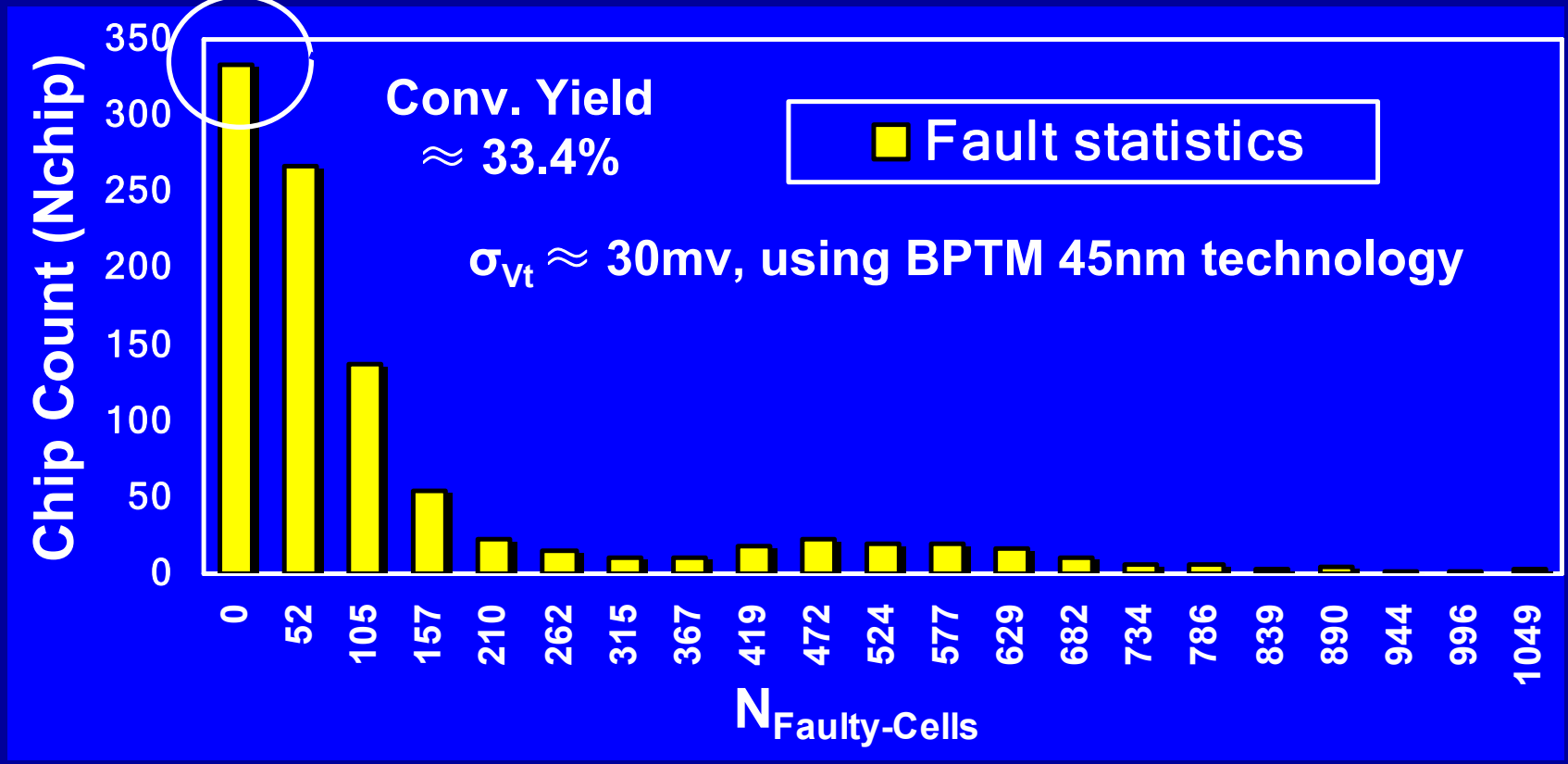
(with 2 blocks in a cache row yield ~ 80%)

Fault Tolerant Capability



- Proposed architecture can handle more number of faulty cells than ECC, as high as 419 faulty cells
- Saves more number of chips than ECC for a given $N_{Faulty\text{-}Cells}$

Process Tolerance: Fault Statistics in 64K Cache

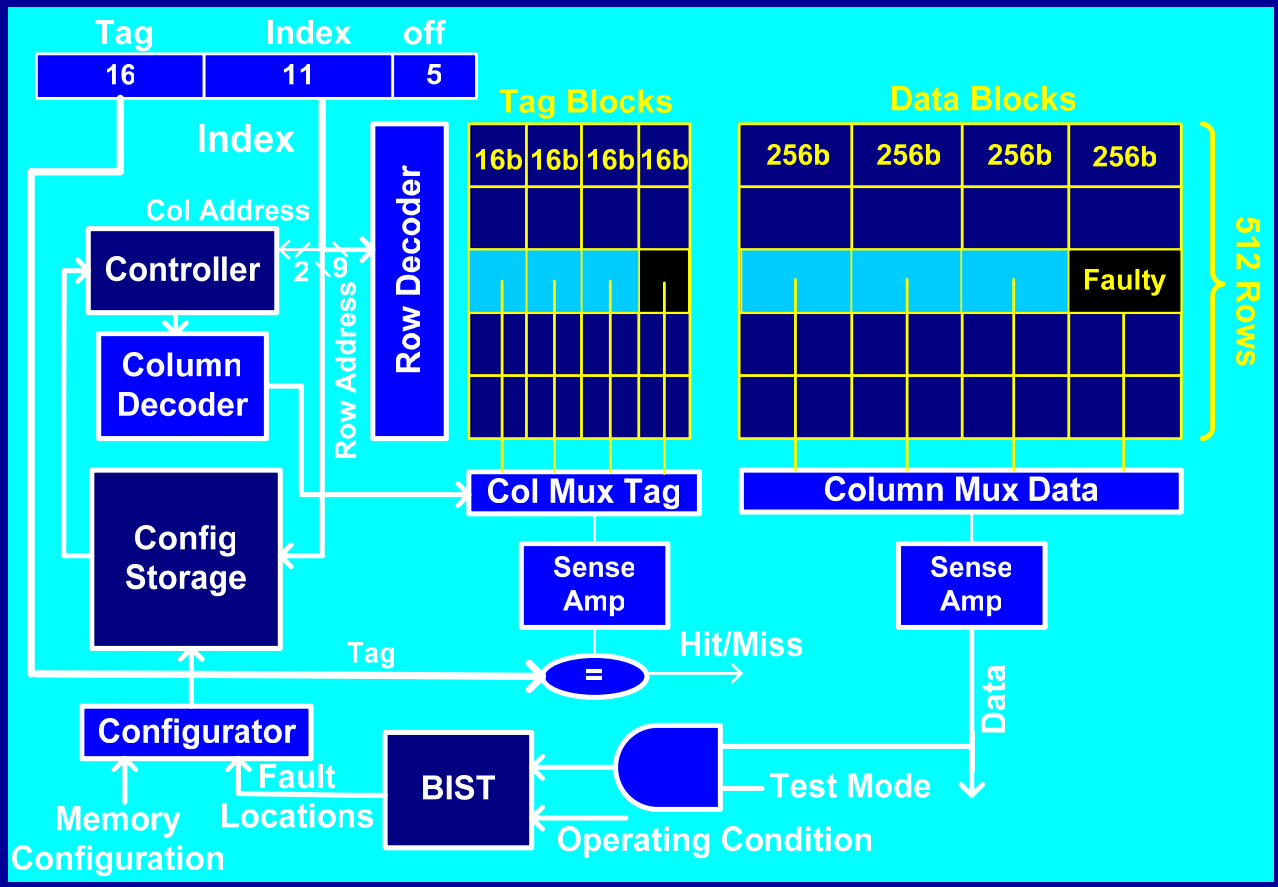


$N_{Faulty\text{-}Cells} = P_{Fault} \; X \; N_{Cells}$ (total number of cells in a cache)

- Conventional 64K cache results in only 33.4% yield

Need a process/fault-tolerant mechanisms to improve the yield in memory

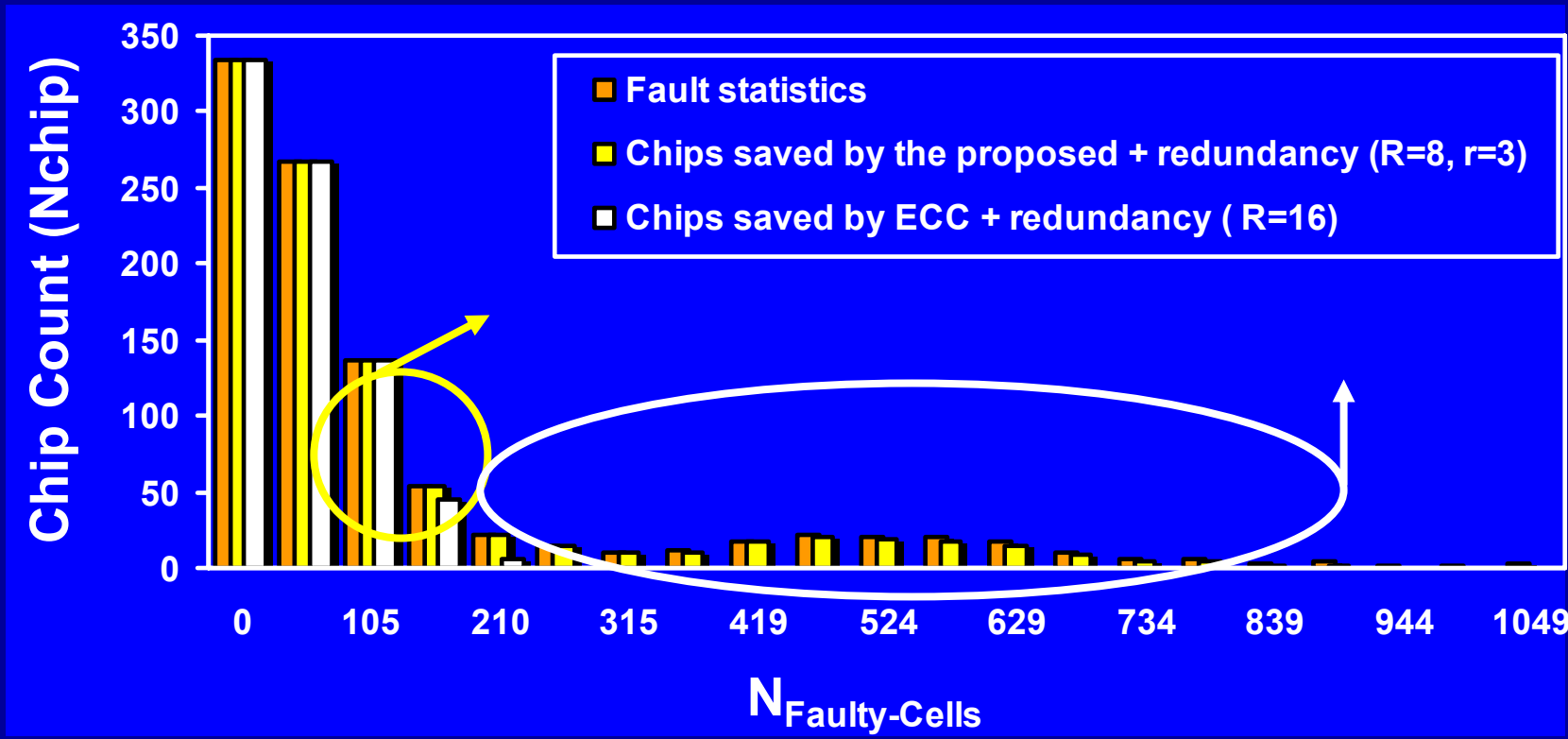
Process-Tolerant Cache Architecture



- BIST detects the faulty blocks
- Config Storage stores the fault information

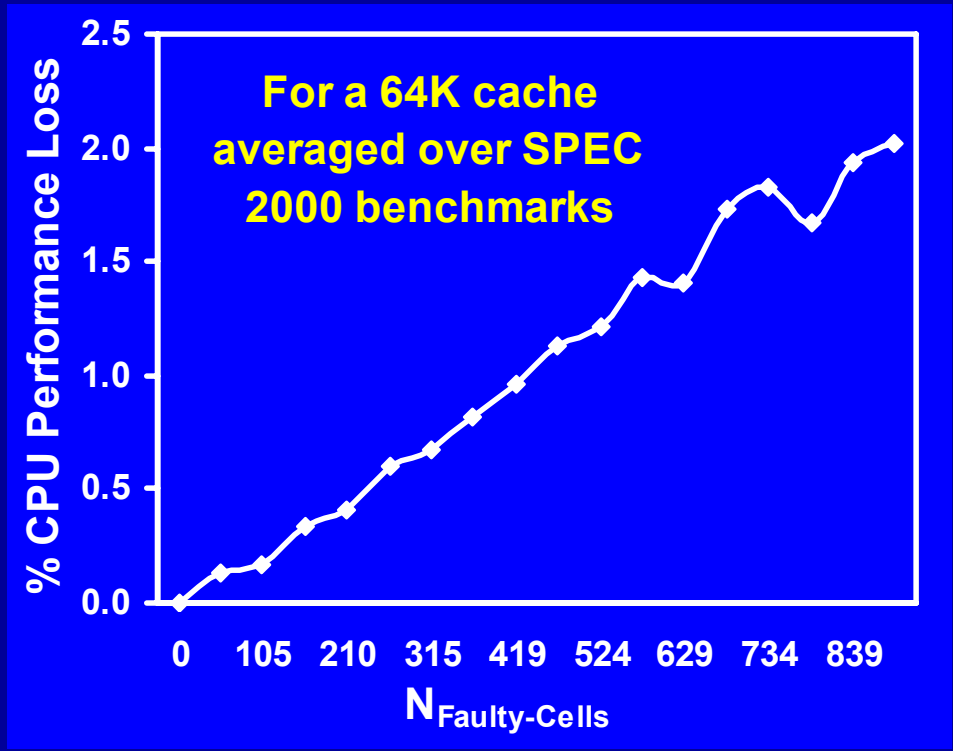
Resize the cache to avoid faulty blocks during regular operation

Fault Tolerant Capability



- Proposed architecture can handle more number of faulty cells than ECC, as high as 890 faulty cells with marginal perf loss

CPU Performance Loss

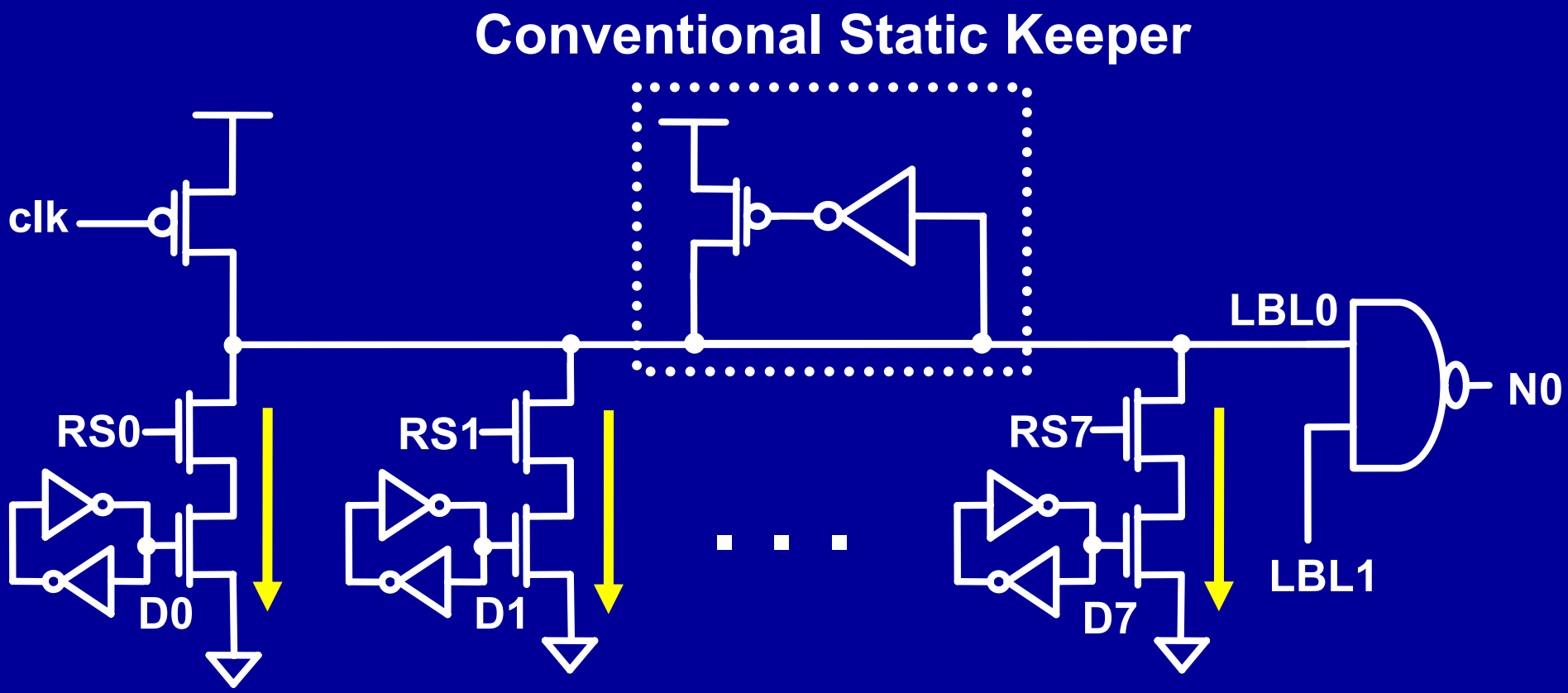


- Increase in miss rate due to downsizing of cache
- Average CPU performance loss over all SPEC 2000 benchmarks for a cache with 890 faulty cells is ~ 2%

Register File: Self-Calibration using Leakage Sensing

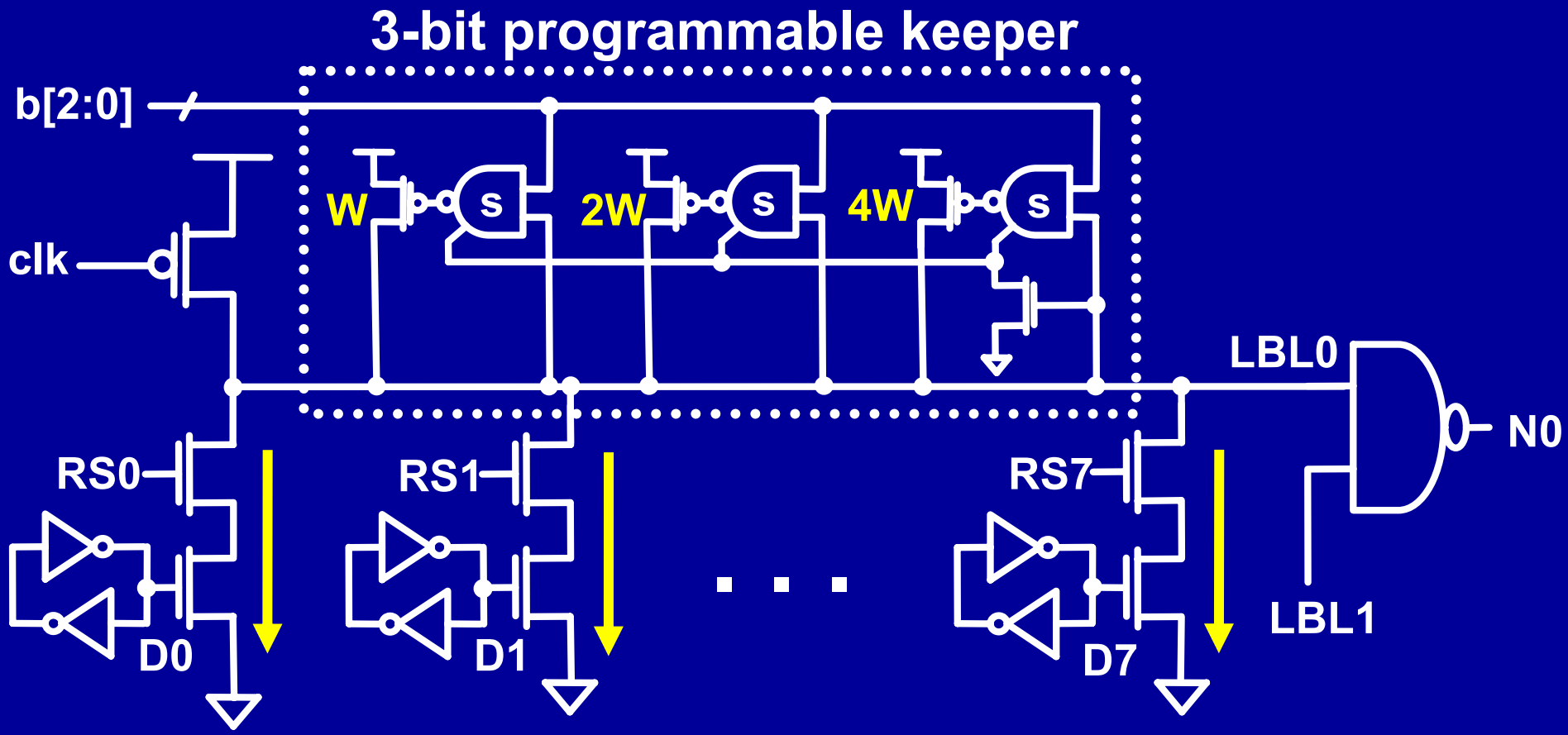
C. Kim, R. Krishnamurthy, & K. Roy

Process Compensating Dynamic Circuit Technology



- Keeper upsizing degrades average performance

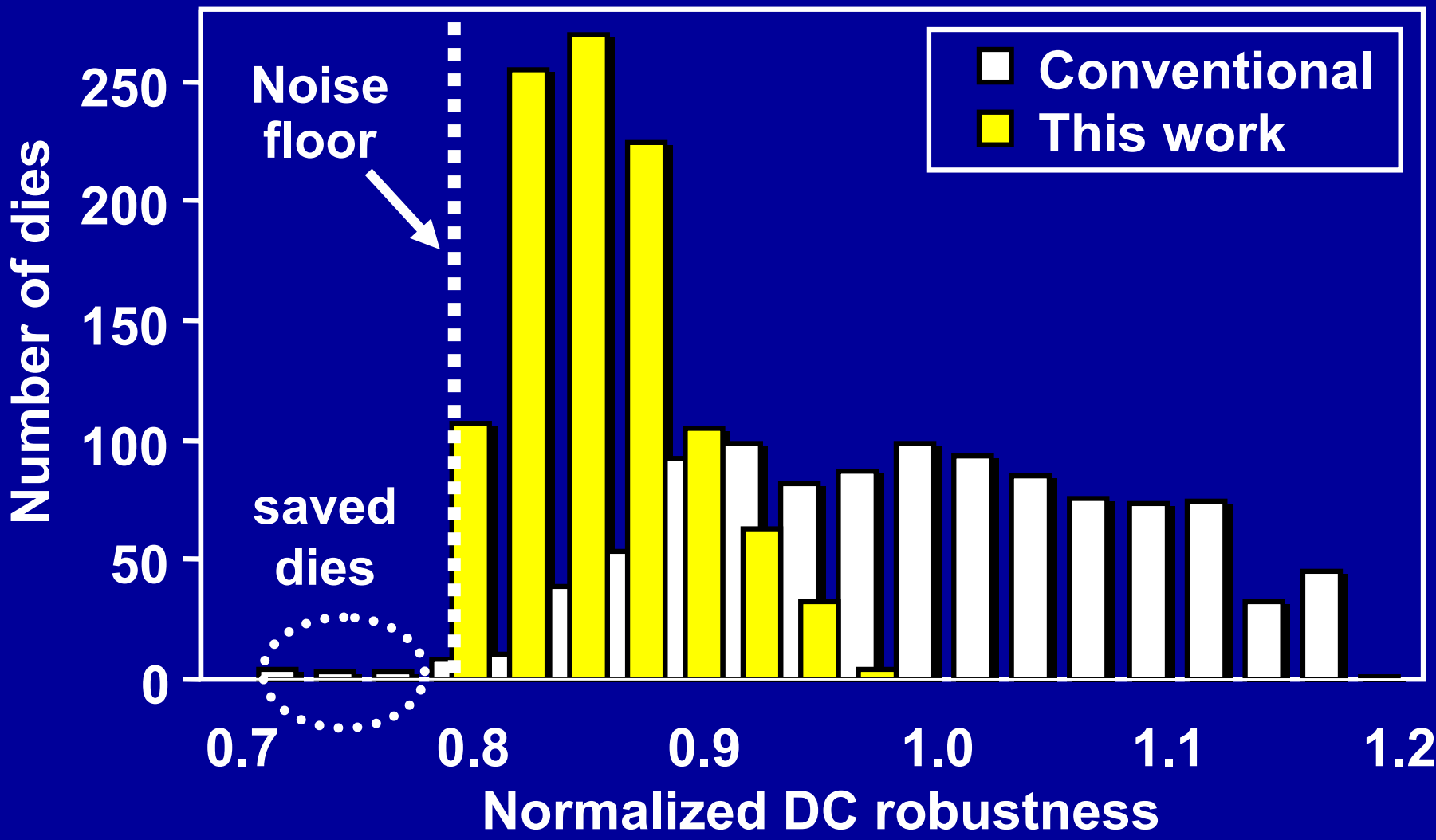
Process Compensating Dynamic Circuit Technology



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

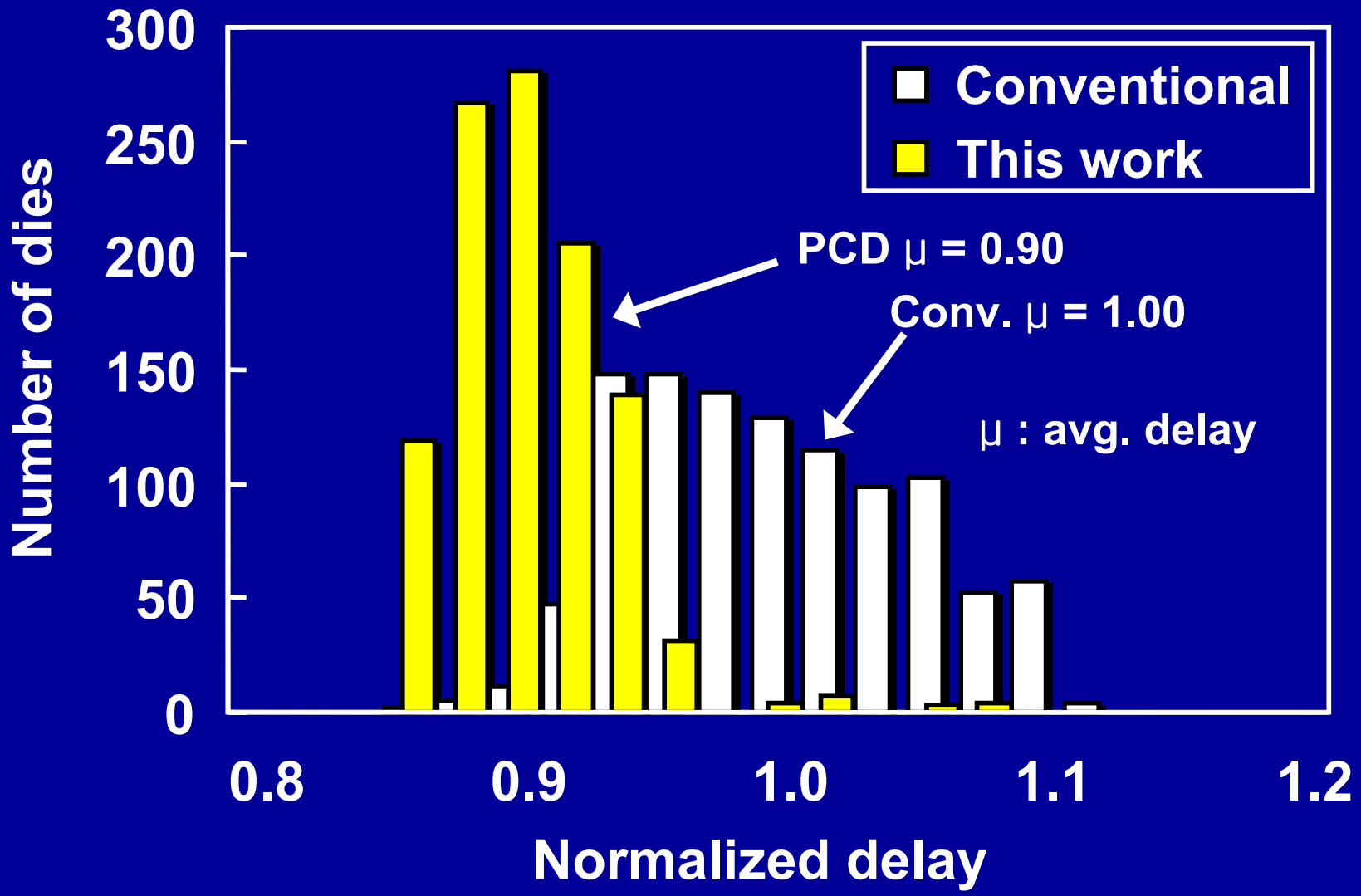
- Opportunistic speedup via keeper downsizing

Robustness Squeeze



- 5X reduction in robustness failing dies

Delay Squeeze

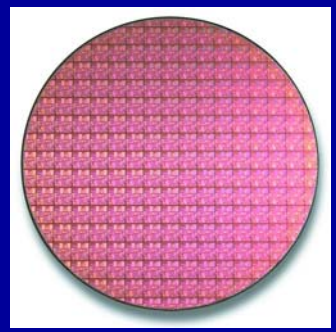


- 10% opportunistic speedup

Self-Contained Process Compensation

Fab

Wafer test



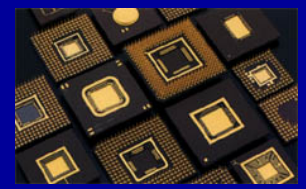
Process detection

Leakage measurement

On-die leakage sensor

Program PCD using fuses

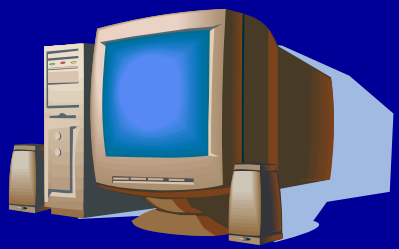
Assembly



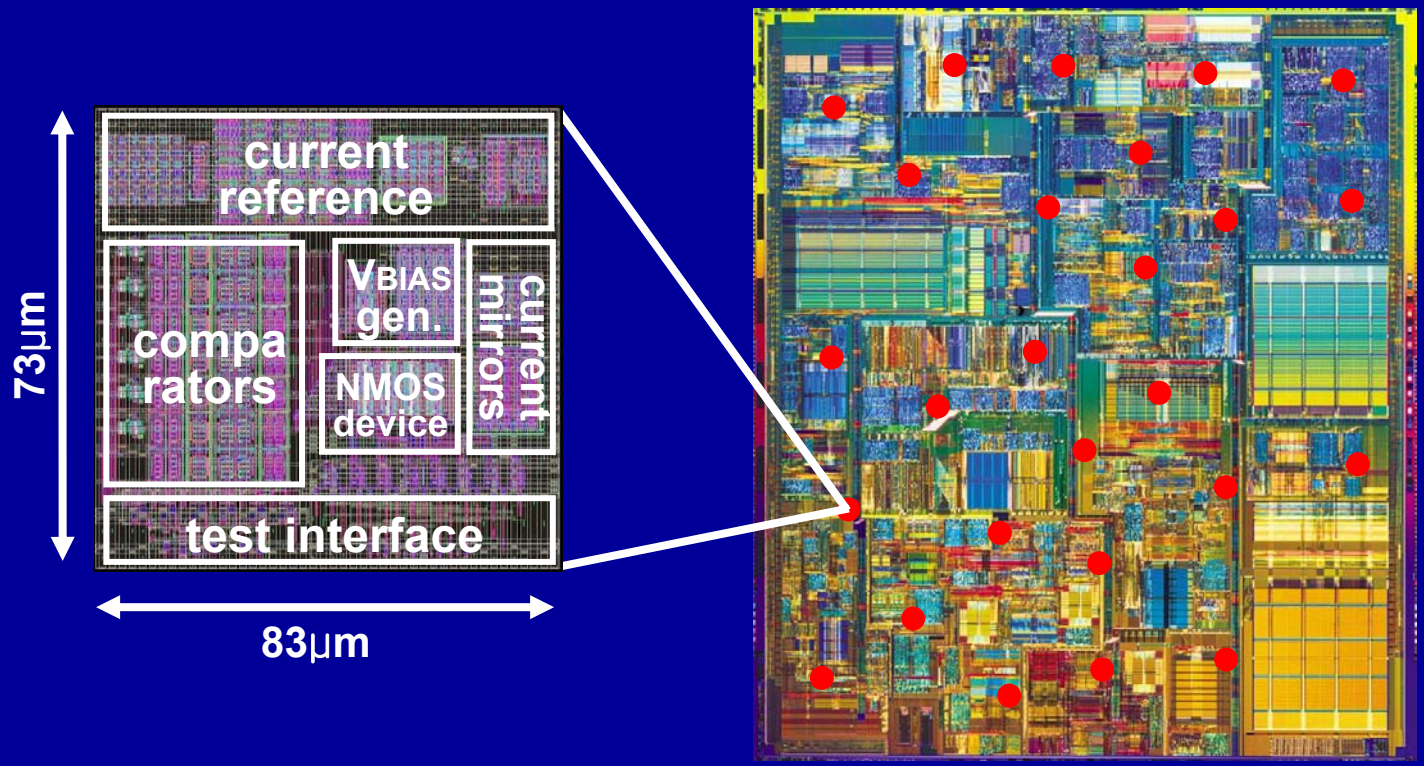
Burn in

Package test

Customer



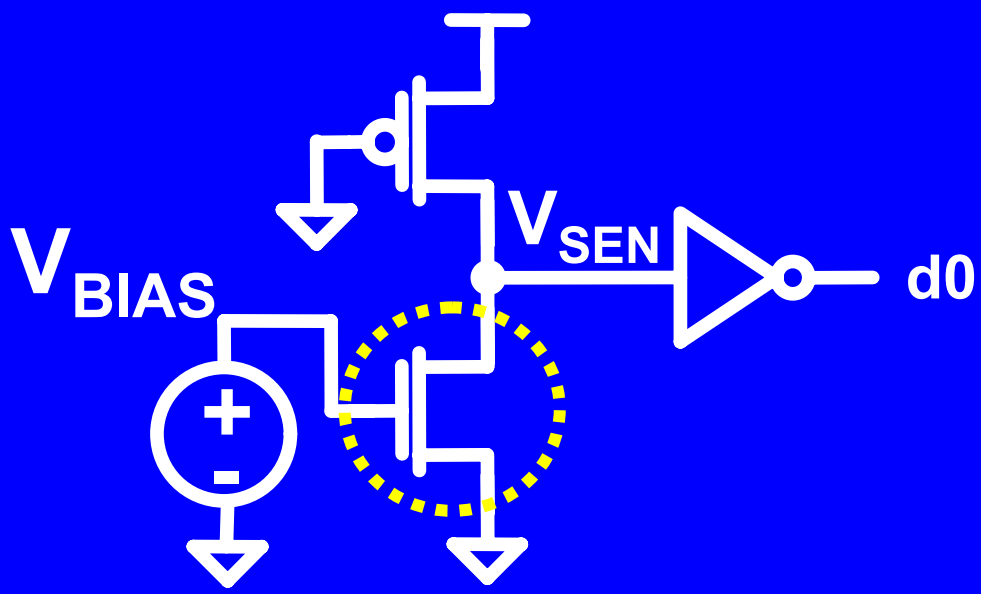
On-Die Leakage Sensor For Measuring Process Variation



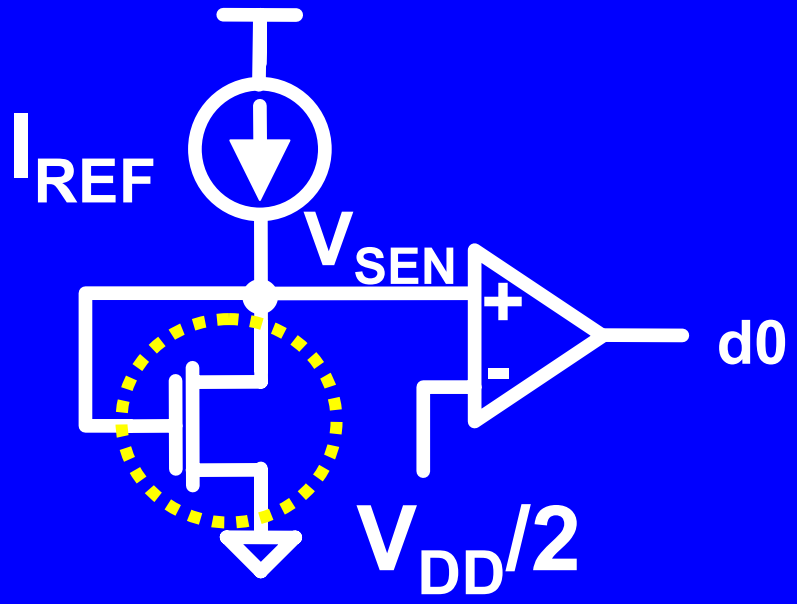
C. Kim et al. , VLSI Circuits Symp. '04

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

Leakage Current Sensing Circuits



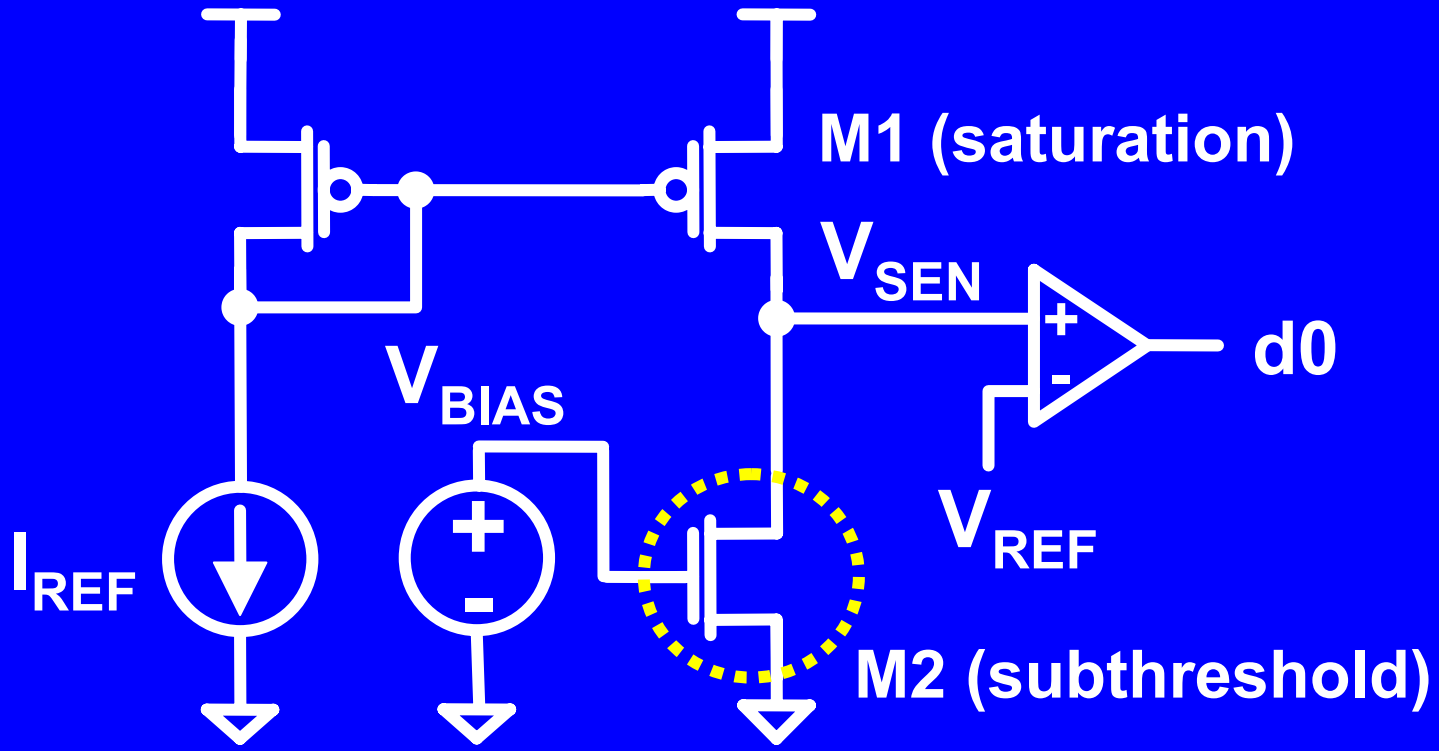
T. Kuroda et al., JSSC, Nov. 1996



M. Griffin et al., JSSC, Nov. 1998

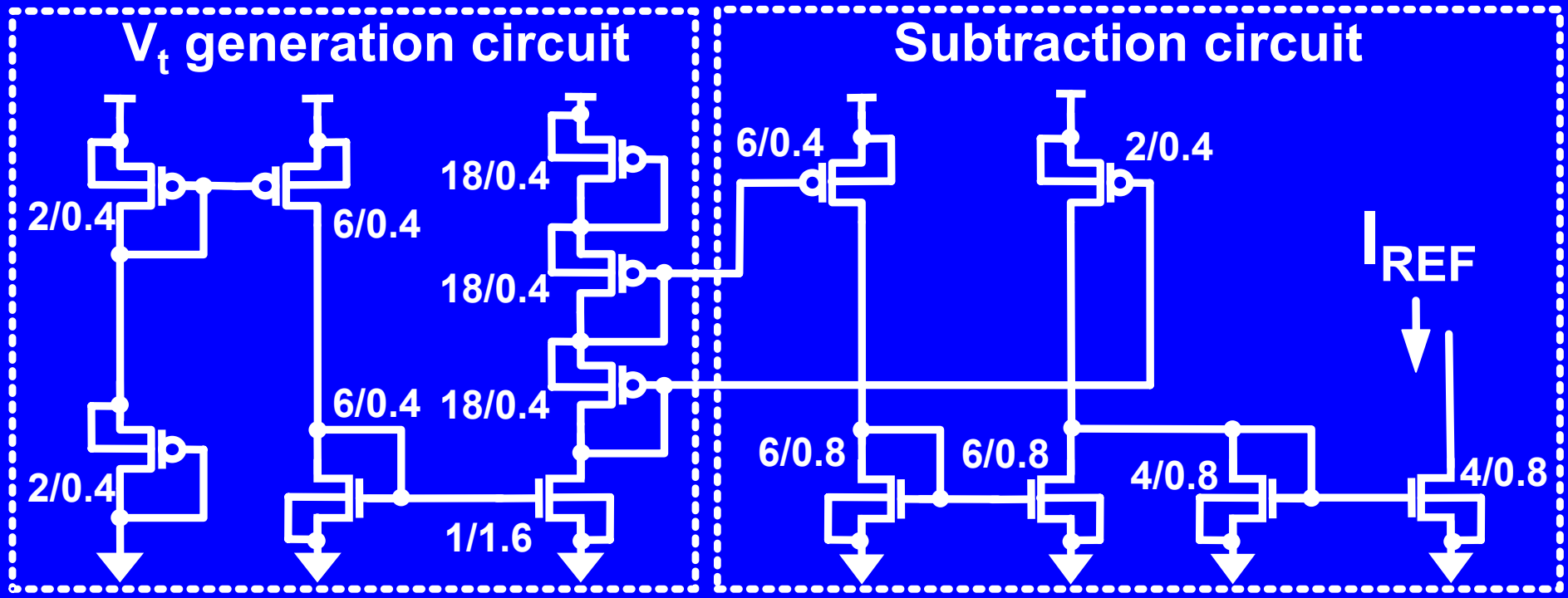
- Susceptible to P/N skew and supply fluctuation
- Large area due to multiple analog bias circuits
- Limited leakage sensing gain

Single Channel Leakage Sensing Circuit



- Basic principle: Drain induced barrier lowering
- Low sensitivity to P/N skew and supply fluctuation

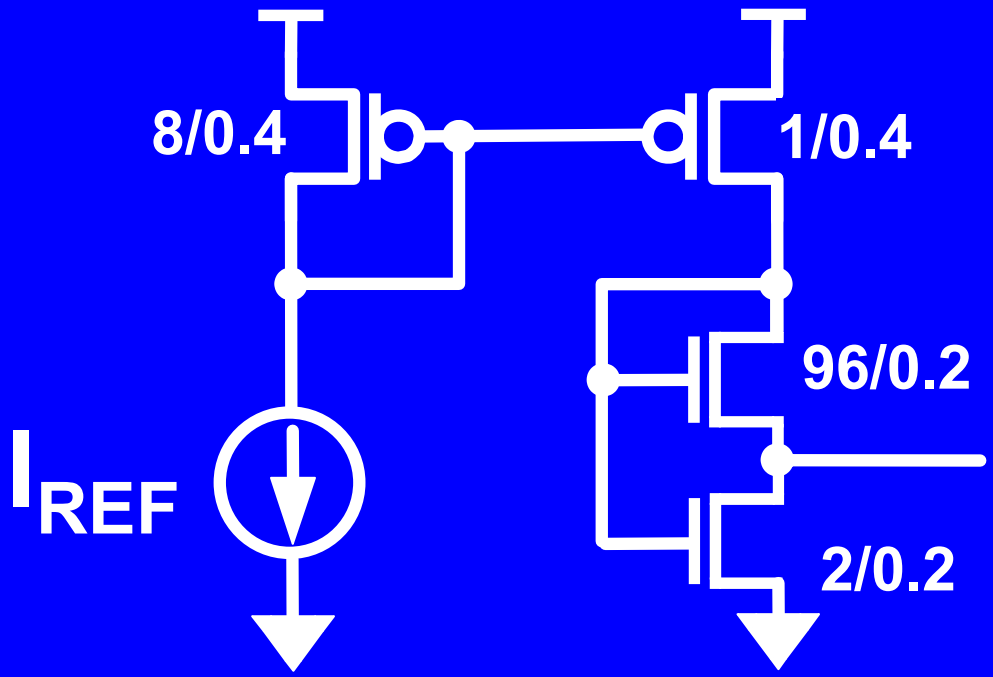
PV Insensitive Current Reference (I_{REF})



S. Narendra et al., VLSI Circuits Symp. 2001

- Sub-1V process, voltage compensated MOS current generation concept
- Reference voltage, external resistor not required
- Scalable, low cost, flexible solution

PV Insensitive Bias Voltage (V_{BIAS})



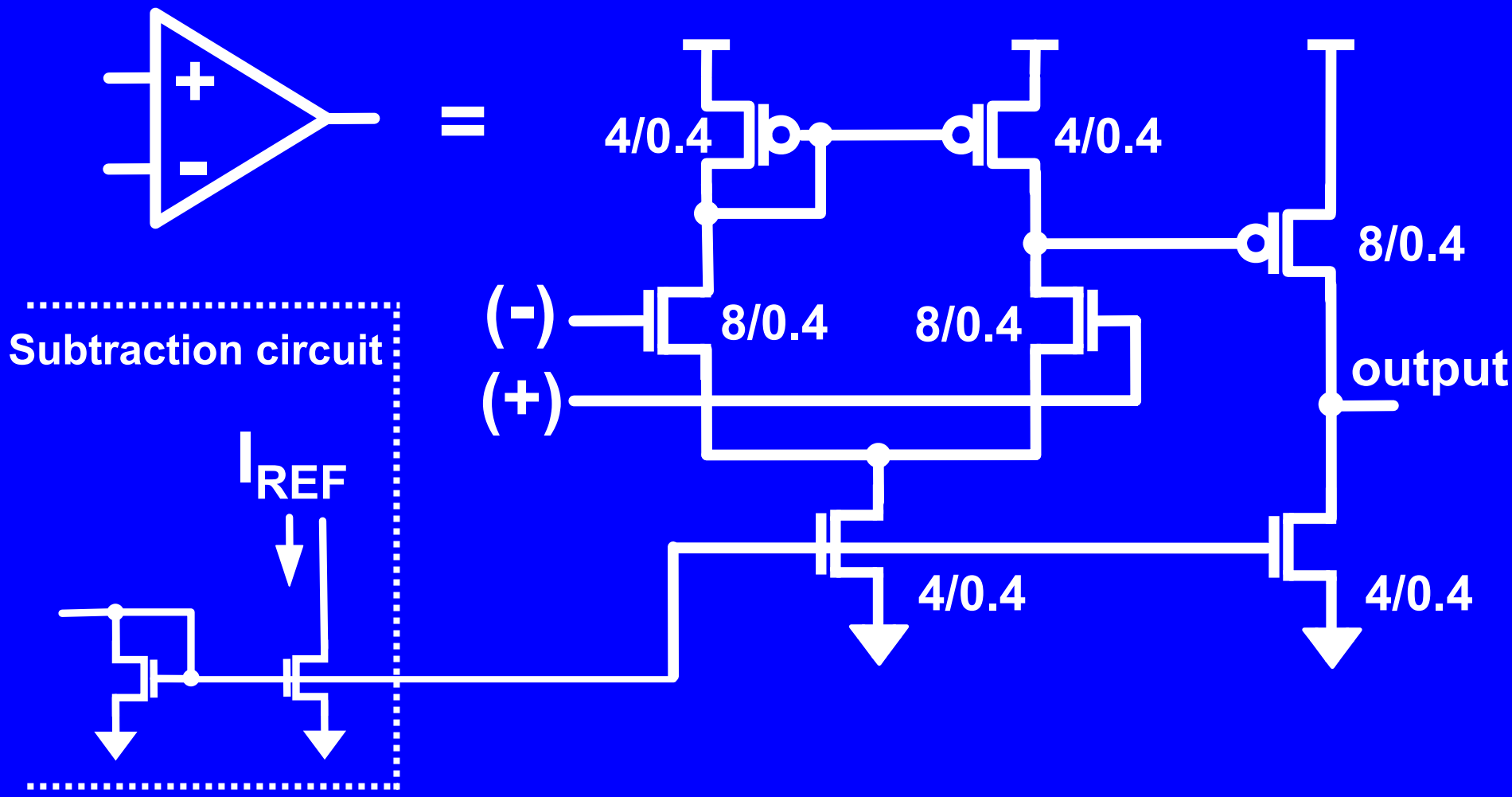
$$V_{BIAS} = \frac{kT}{q} \log\left(\frac{W_1}{W_2}\right)$$

(W_1=96μm, W_2=2μm)

E. Vittoz et al., JSSC, June 1979

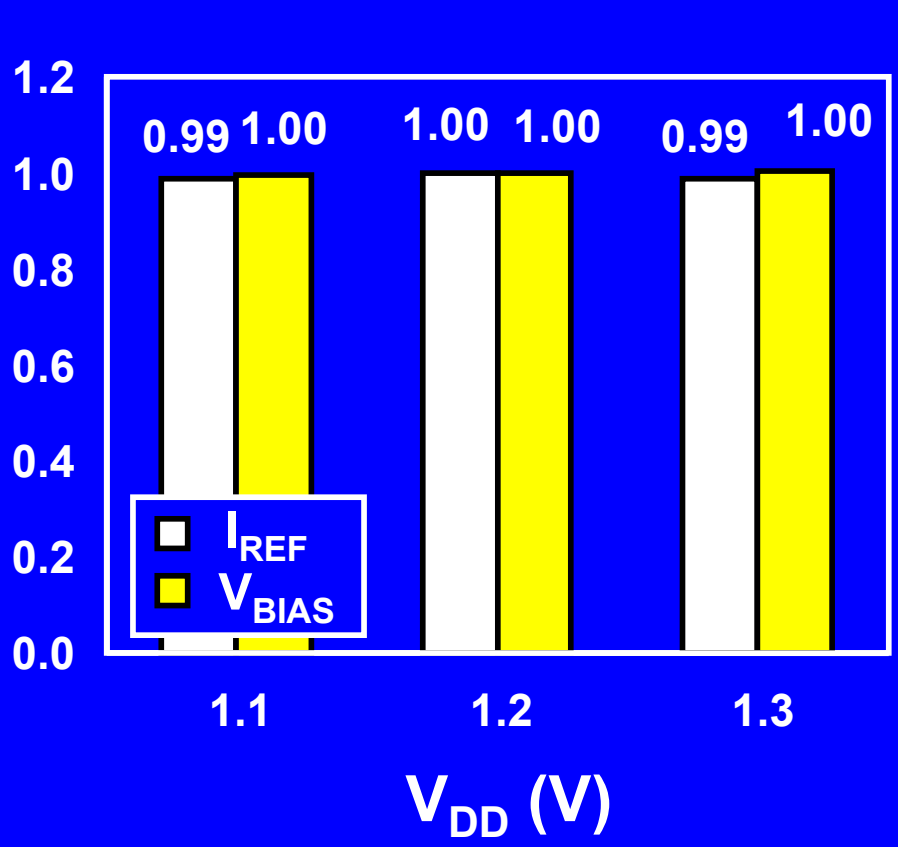
- PTAT containing no resistive dividers
- Based on weak inversion MOS characteristics
- Desired output voltage achieved via sizing

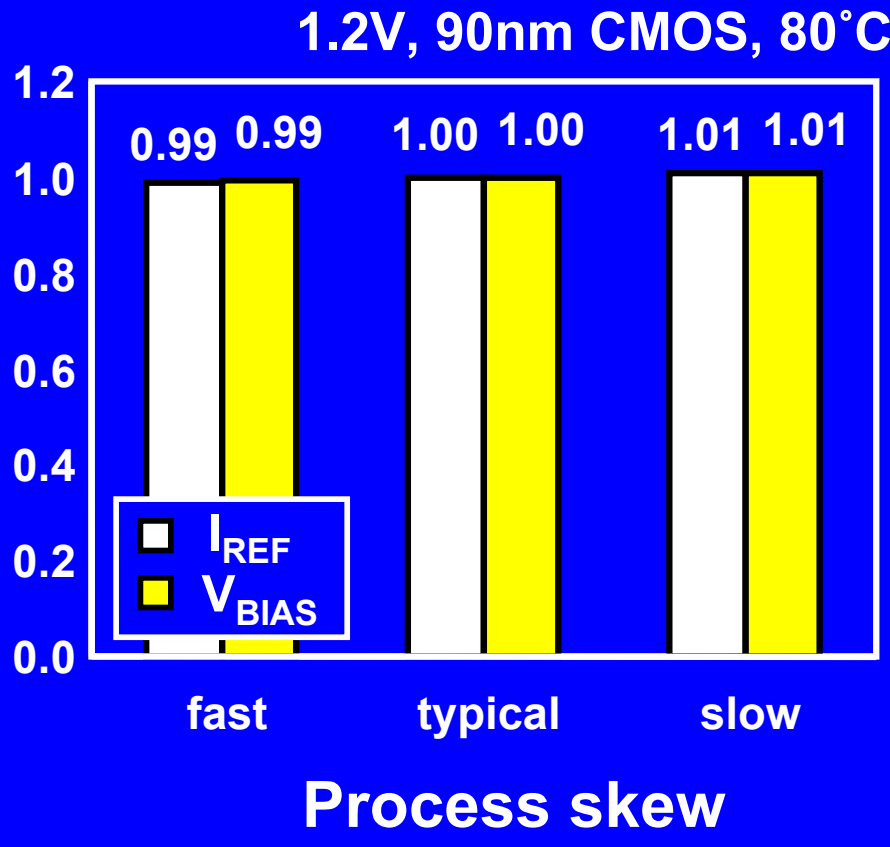
Comparator



- 2-stage differential amplifier
- Already designed I_{REF} is used for bias current

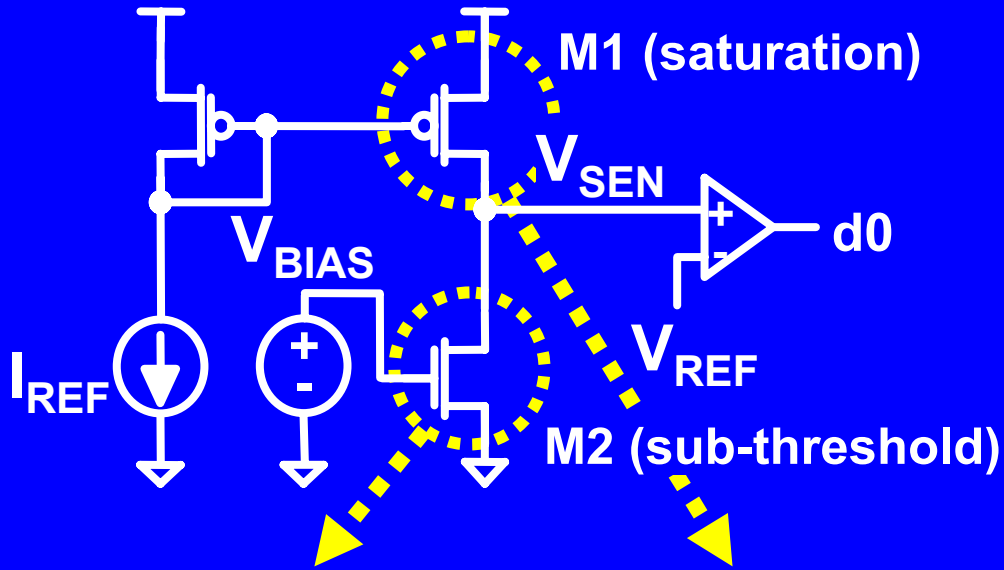
PV Sensitivity of Designed I_{REF}, V_{BIAS}

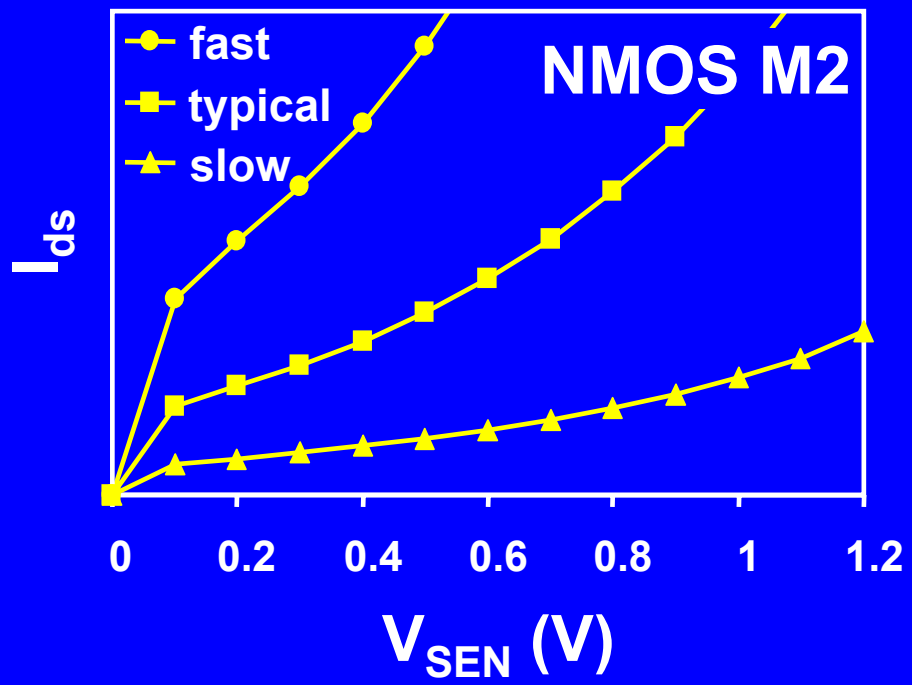


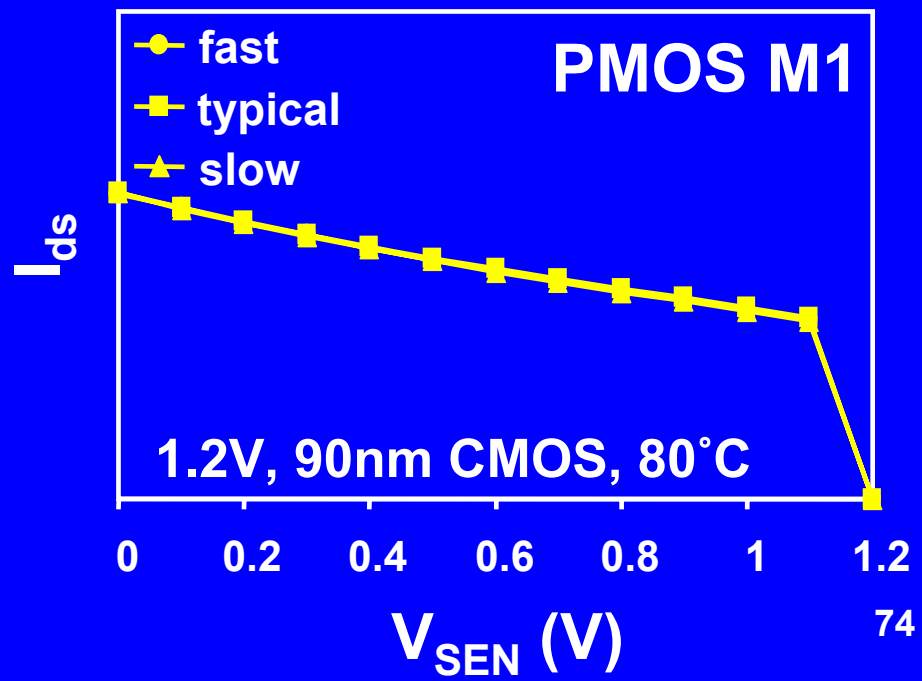


- I_{REF} variation < 4%, V_{BIAS} variation < 2%
- Under realistic process skews, ±100mV supply voltage fluctuations

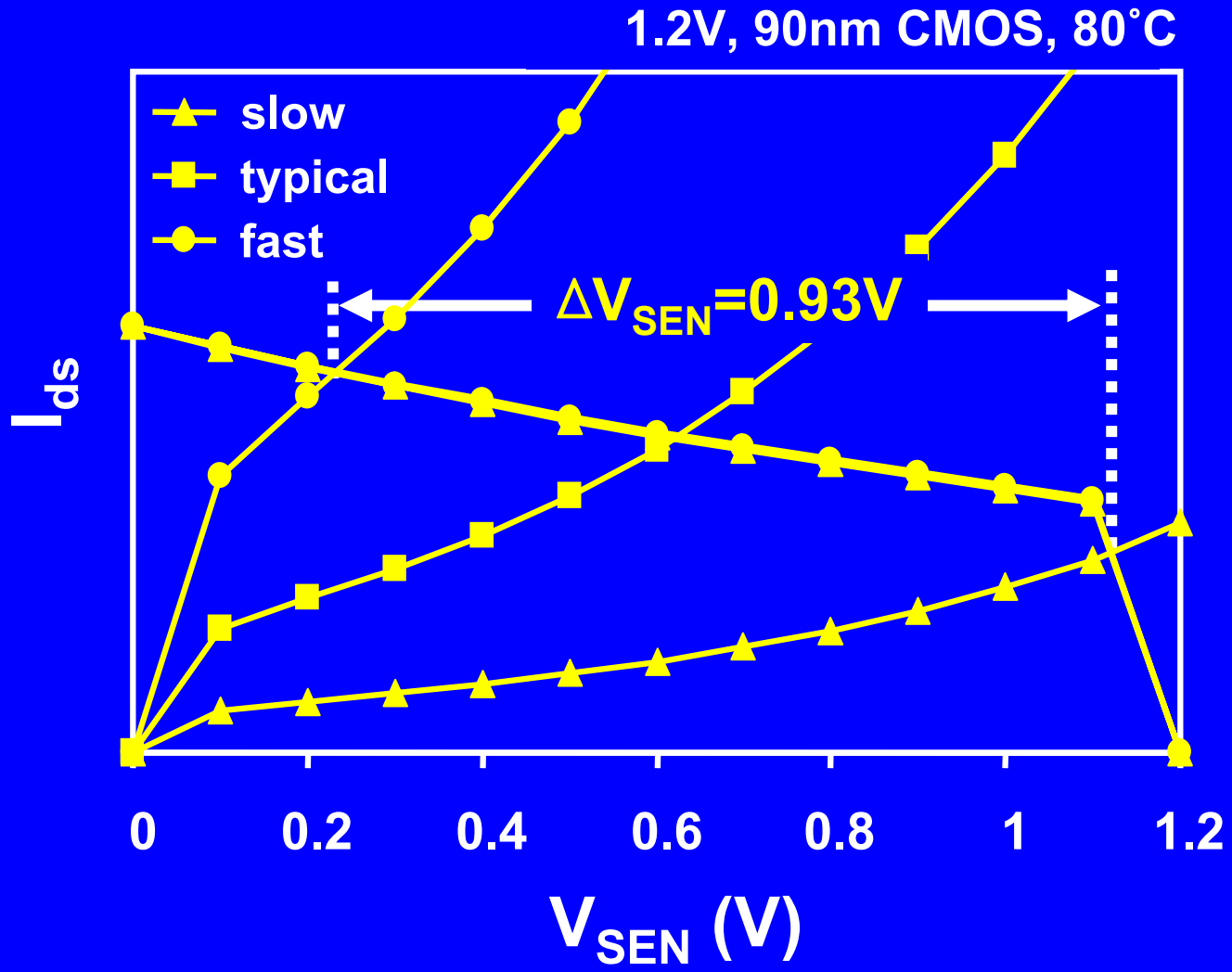
Proposed Leakage Current Sensing





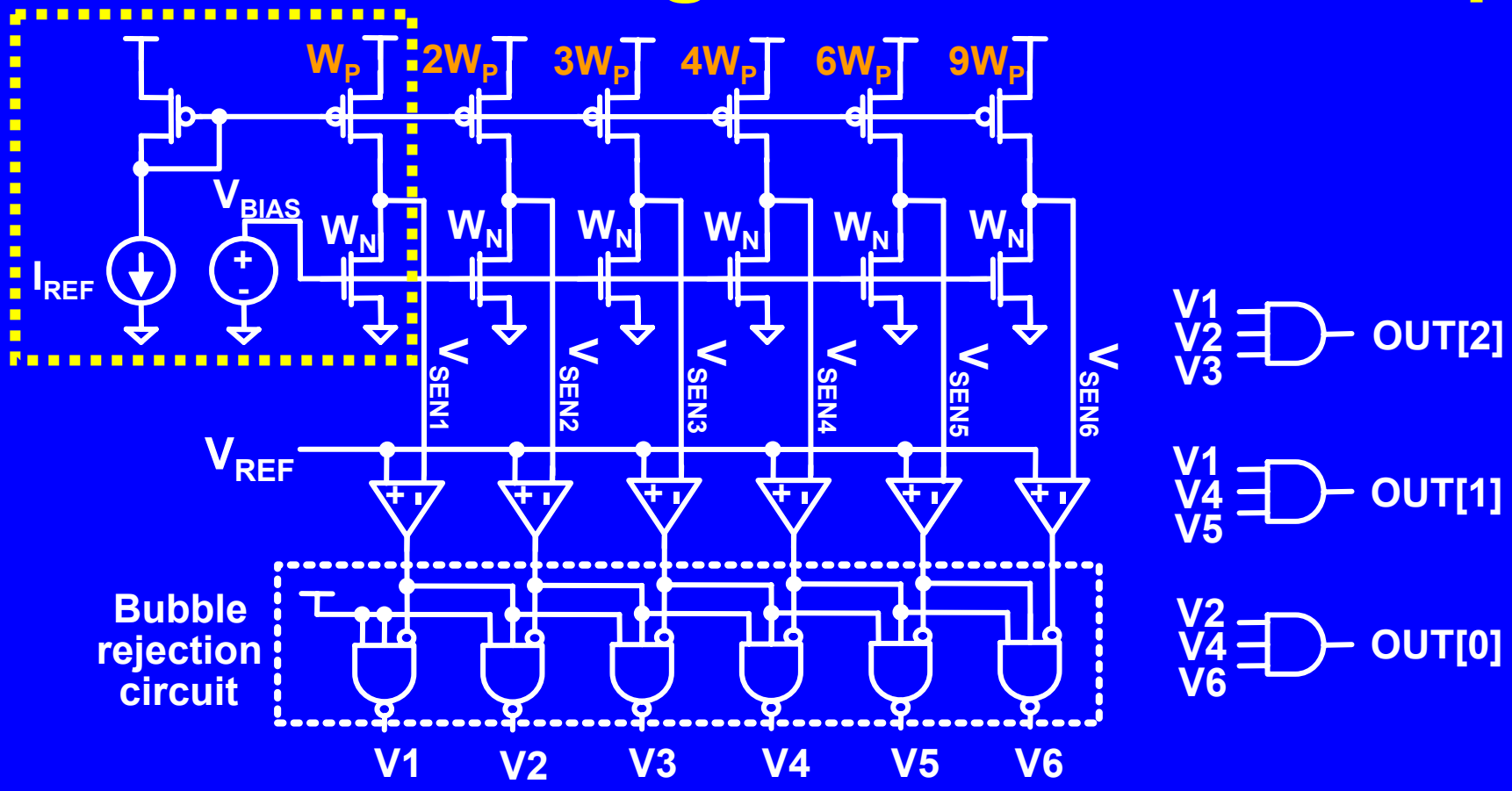


Superimposed I-V Curves



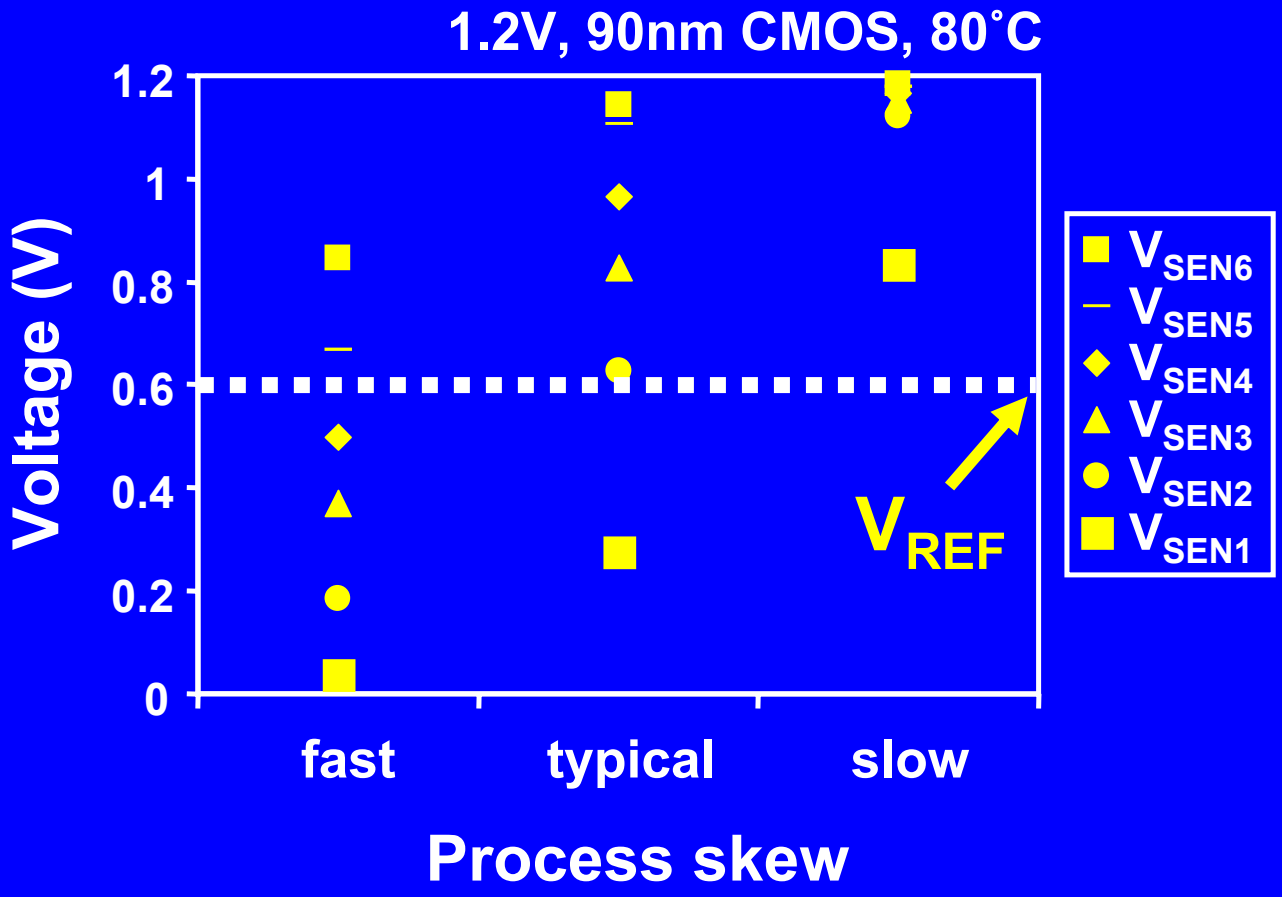
- 1.9-10.2X higher V_{SEN} swing than prior-art
- Process-voltage insensitive design

6-Channel Leakage Sensor Test Chip



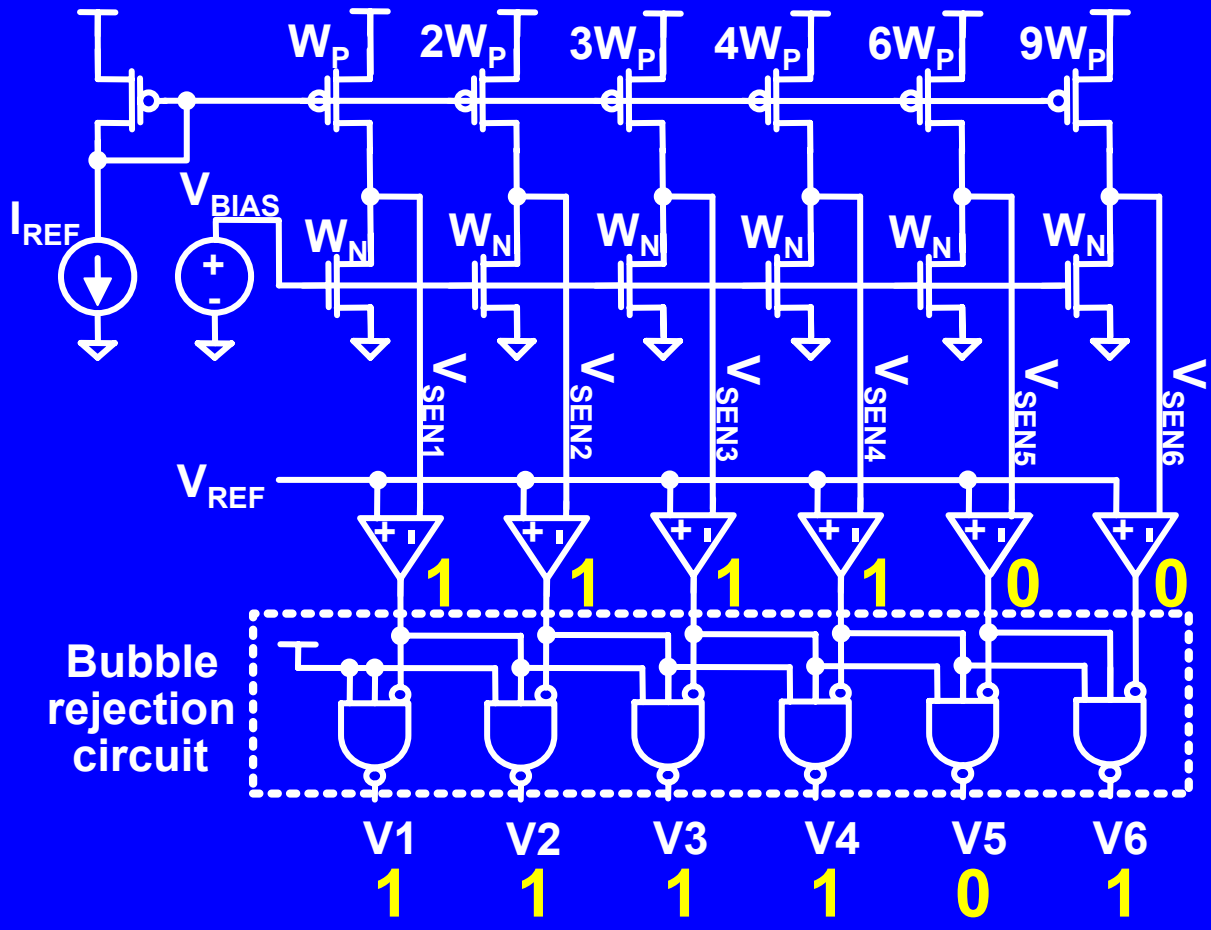
- Incremental mirroring ratio for multi-bit resolution leakage sensing
- Shared bias generators → compact design
- Process-voltage insensitive I_{REF}, V_{BIAS} gen.

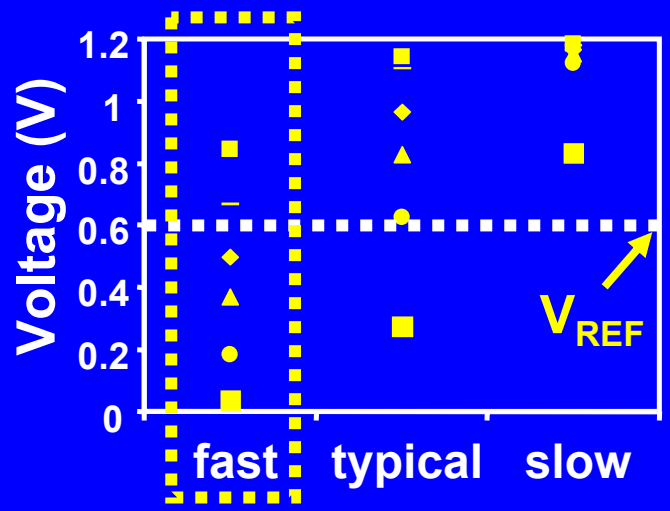
Multi-Bit Resolution Leakage Sensing

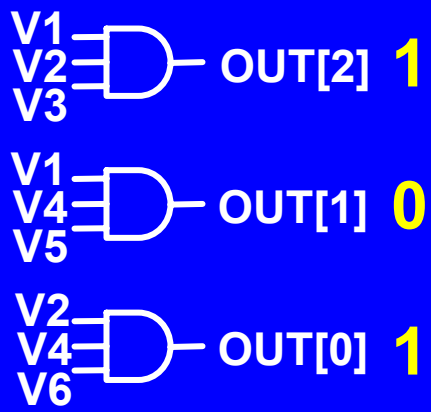


- Leakage level determined by comparing V_{SEN1} through V_{SEN6} with V_{REF}
- 6-channel leakage sensor gives 7 level resolution

Example: Operation at Fast Process Corner

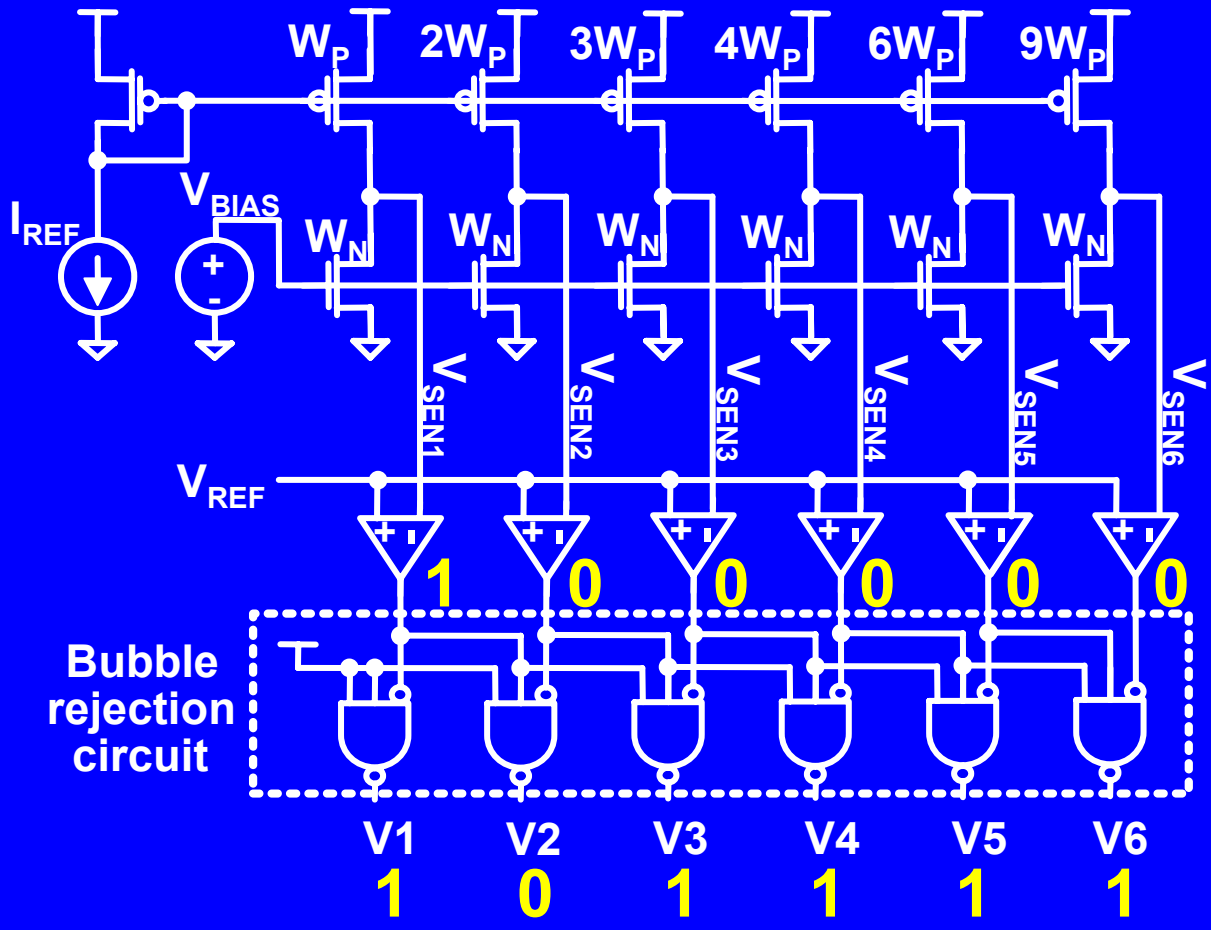


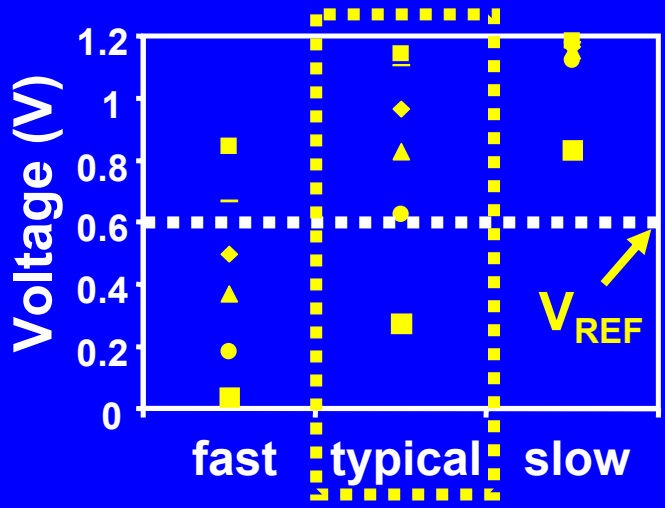


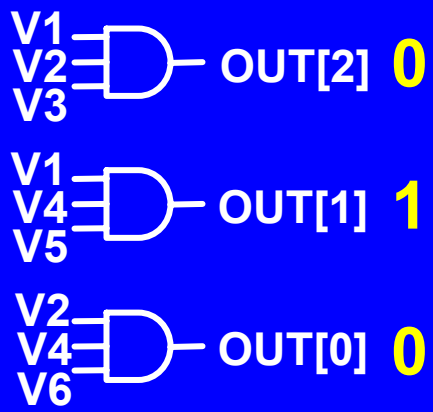


- Fast corner: output code ‘101’

Example: Operation at Typical Process Corner

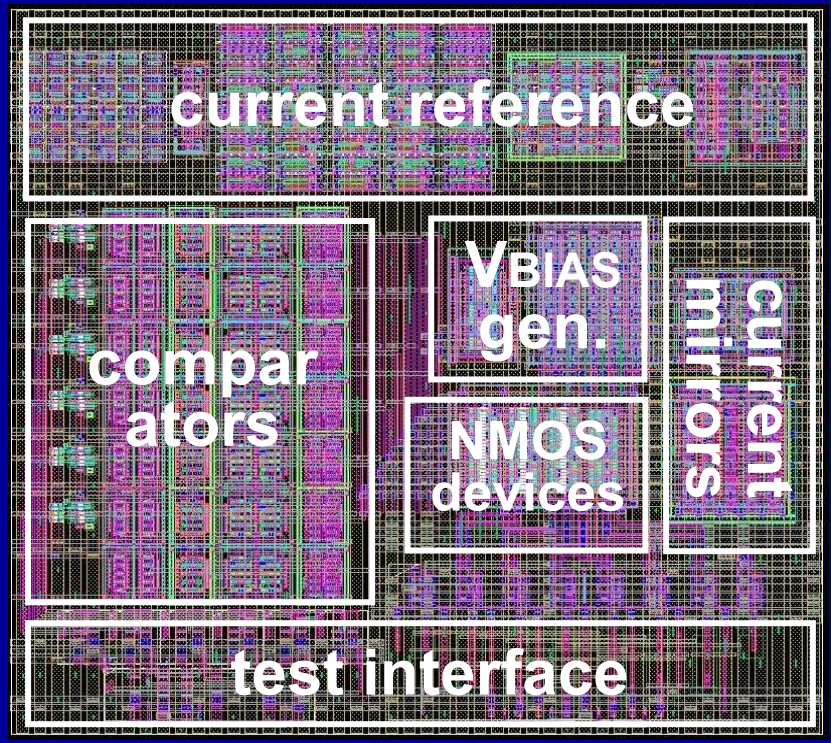






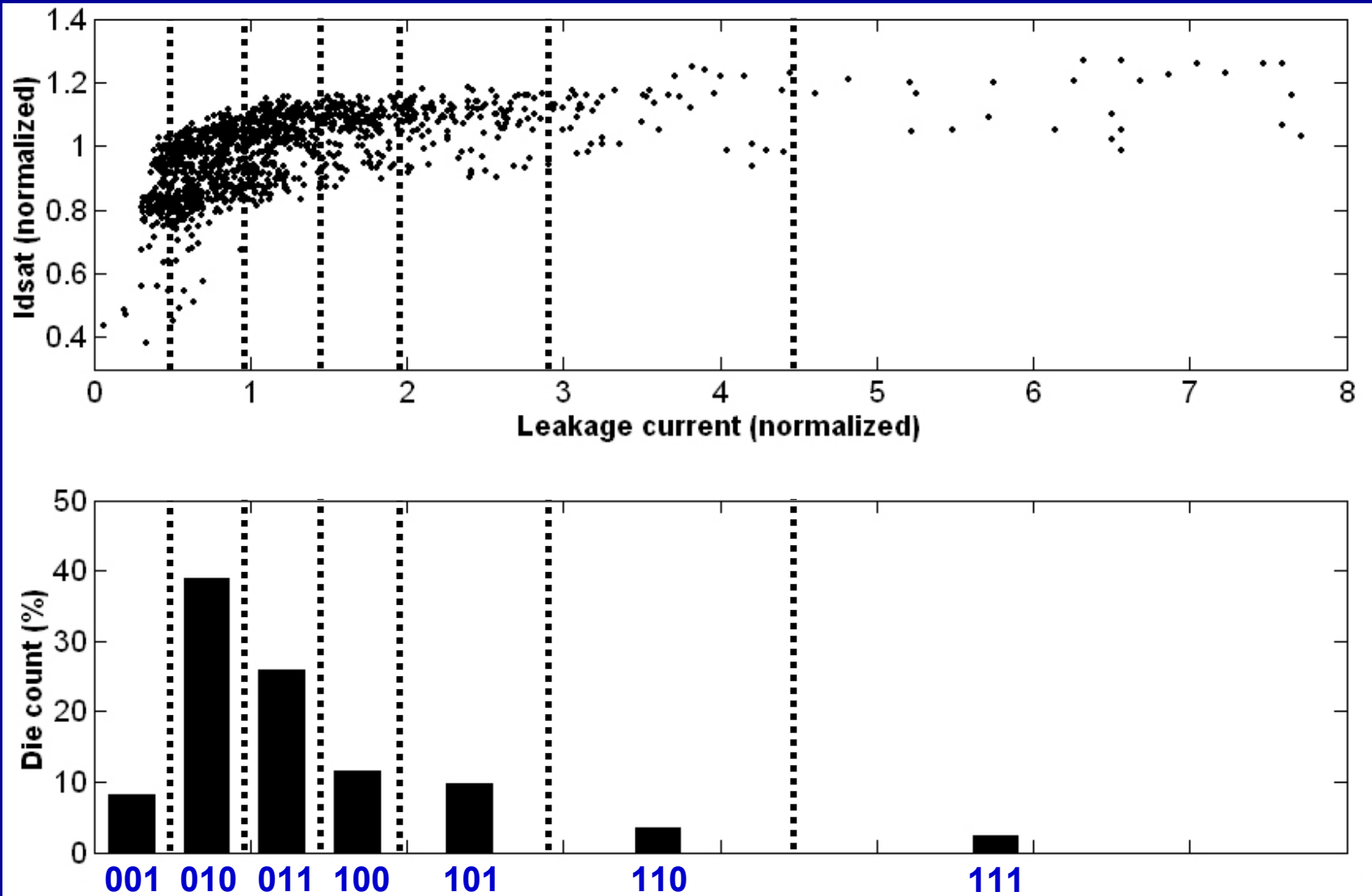
- Typical corner: output code ‘010’

On-Die Leakage Sensor Test Chip



Technology	90nm dual Vt CMOS
V_{DD}	1.2V
Resolution	7 levels
Power consumption	0.66 mW @80C°
Dimensions	83 X 73 μm^2

Leakage Binning Results



Output codes from leakage sensor

Conclusion

- Statistical Failure Analysis Helps Enhance Yield
- Post Silicon Tuning/Calibration is Becoming Promising for Si Nano systems
- Built-In Leakage/Delay Sensors Provide Information on Intra-Die Process Variations