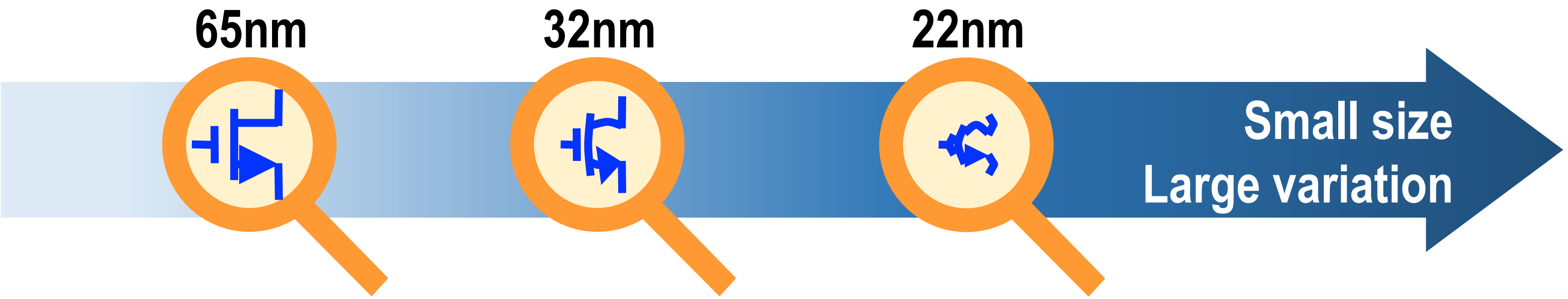
Multi-corner Parametric Yield Estimation via Bayesian Inference on Bernoulli Distribution with Conjugate Prior

Jiahe Shi[1], Zhengqi Gao[1], Jun Tao[1], Yangfeng Su[2], Dian Zhou[3] and Xuan Zeng[1]
1 ASIC & System State Key Lab, School of Microelectronics, Fudan University, Shanghai, China,
2 School of Mathematical Sciences, Fudan University, Shanghai, China,
3 Dept. of EE, University of Texas at Dallas, Dallas, USA

2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

Parametric Yield Estimation

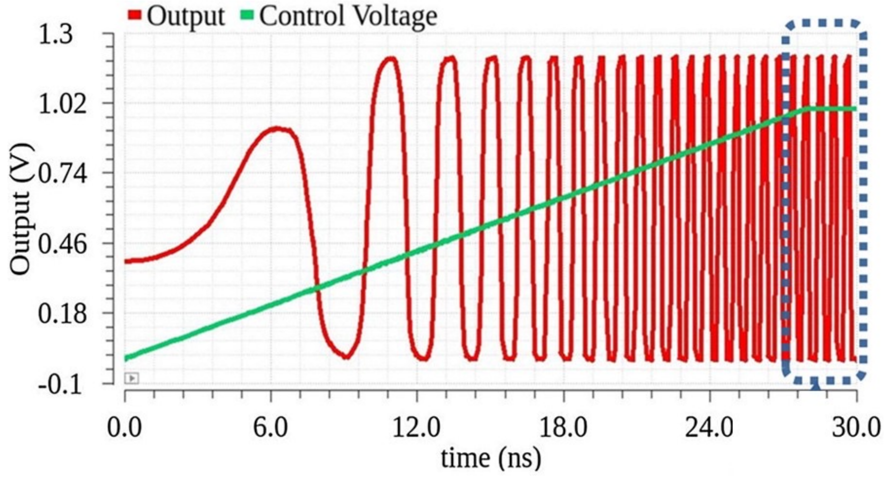


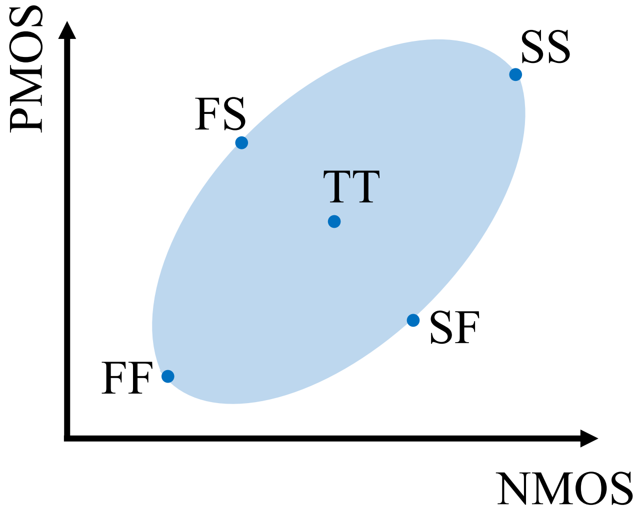
Circuit performances uncertainty

Guarantee the robustness of circuit designs

- **Parametric Yield Estimation methods**
  - Statistical approaches
    - Estimate the PDF of the performance of interests (PoIs) from a large number of transistor-level simulations
  - Model-based methods
    - Approximate the PoIs by analytical functions of process variations
    - Estimate the PDFs of PoIs via numerical methods

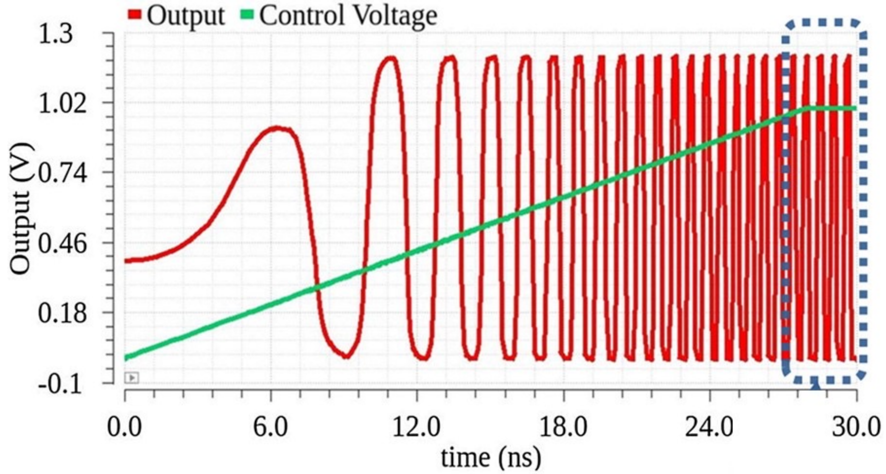
Parametric Yield Estimation

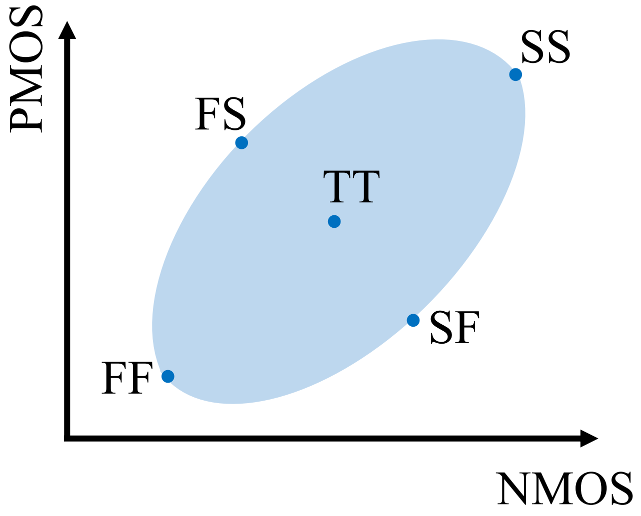




- **CHALLENGES**
  - **Measuring the exact value of a continuous PoI is expensive**
    - VCO
  - **Circuit Performances over all process, voltage and temperature (PVT) corners are correlated**
    - Different process corners

Parametric Yield Estimation





- **CHALLENGES**

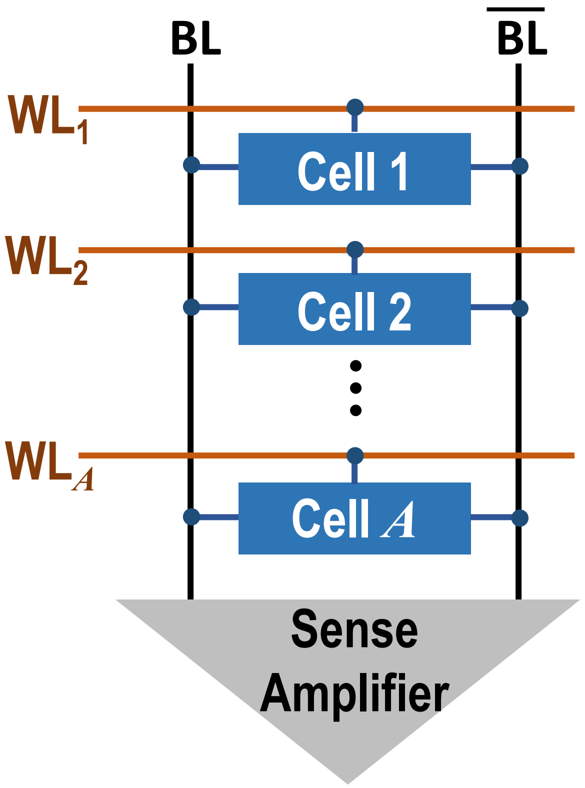
- **Measuring the exact value of a continuous PoI is expensive**
  - VCO

Model the circuit output as a Bernoulli random variable

- **Circuit Performances over all process, voltage and temperature (PVT) corners are correlated**
  - Different process corners

Encode the correlation of yields over all corners

Multi-corner Yield Estimation



Simplified SRAM array with A cells

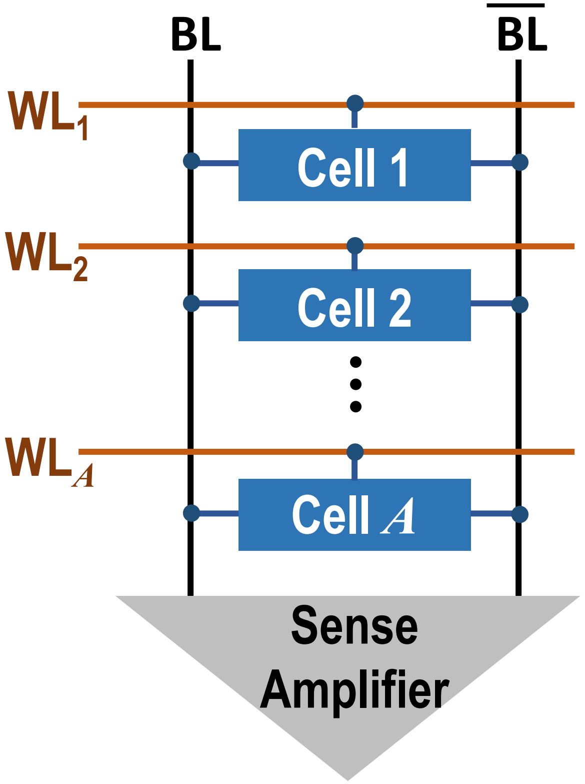
- **Example: Access operation**

$$x = \begin{cases} 1 & \text{if success} \\ 0 & \text{if fail} \end{cases}$$

$$p\left(x \middle| \beta_k\right) = \beta_k^x \cdot \left(1 - \beta_k\right)^{1\text{-}x}$$

x: Output of the binary testing
β_k: Parametric yield at the k-th corner

Multi-corner Yield Estimation



Simplified SRAM array with A cells

- **Example: Access operation**

$$x = \begin{cases} 1 & \text{if success} \\ 0 & \text{if fail} \end{cases}$$

$$p\left(x \middle| \beta_k\right) = \beta_k^x \cdot \left(1 - \beta_k\right)^{1-x}$$

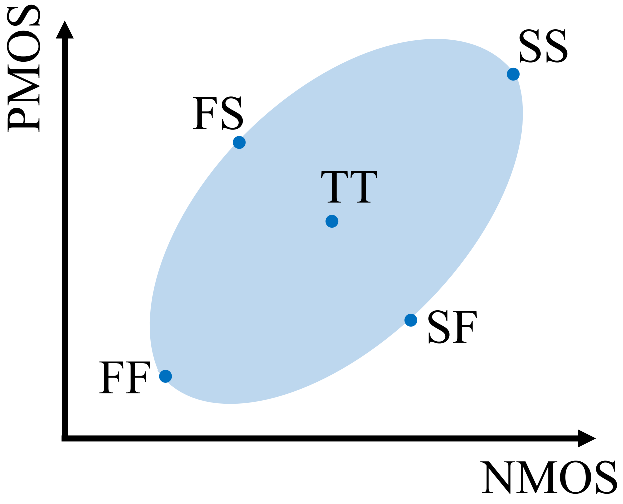
- **Likelihood**

$$p\left(\mathrm{D} \middle| \boldsymbol{\beta}\right) = \prod_{n=1}^{N} \prod_{k=1}^{K} \beta_k^{x_k^n} \cdot \left(1 - \beta_k\right)^{1-x_k^n}$$

$$= \prod_{k=1}^{K} \beta_k^{M_k} \cdot \left(1 - \beta_k\right)^{N-M_k}$$

where, $M_k = \sum_{n=1}^{N} x_k^n$

x: Output of the binary testing
β_k: Parametric yield at the k-th corner
D: Training dataset over all corners
N: Number of data at one corner
K: Number of corners
$\boldsymbol{\beta}$: $\boldsymbol{\beta} = [\beta_1\ \beta_2 \cdots \beta_K]^T$



Multi-corner Yield Estimation

- **Problem definition:**

$$p(\boldsymbol{\beta}|\mathrm{D}) \propto p(\boldsymbol{\beta}) \cdot p(\mathrm{D}|\boldsymbol{\beta})$$

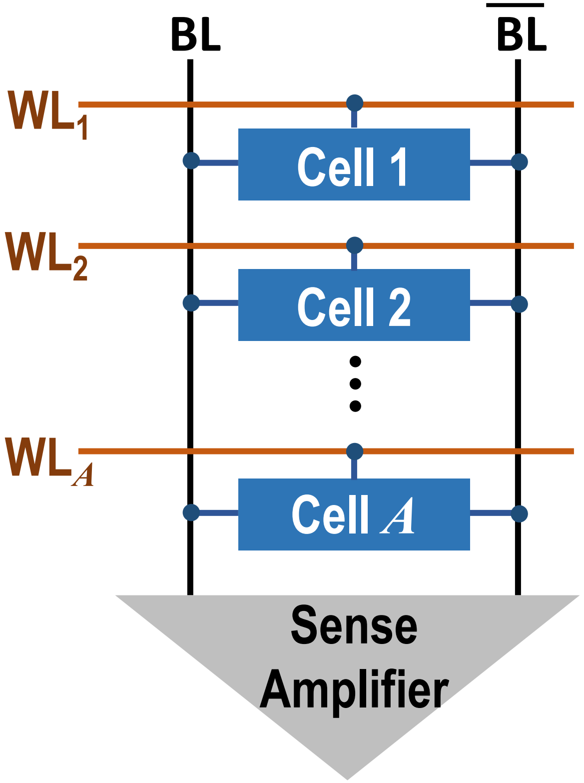
$$\max_{\boldsymbol{\beta}} \quad p(\boldsymbol{\beta}|\mathrm{D})$$

$p(\boldsymbol{\beta}|D)$: Posterior distribution
$p(\boldsymbol{\beta})$: Prior distribution
$p(D|\boldsymbol{\beta})$: Likelihood

- **Bayesian Inference method based on Bernoulli distribution (BI-BD) [TCAD 2019]**
  - **Models the circuit output as a Bernoulli random variable**
  - **Adopts a multivariate Gaussian distribution as prior distribution**
  - **Calculate the multi-corner yields via posterior distribution approximation**

Can not analytically derive the exact posterior but approximate it via an iterative method

Bayesian Inference



Simplified SRAM array with A cells

- **Likelihood function**

$$p(\mathrm{D}|\boldsymbol{\beta})=\prod_{n=1}^{N}\prod_{k=1}^{K}\beta_k^{x_k^n}\cdot(1-\beta_k)^{1-x_k^n}$$

$$=\prod_{k=1}^{K}\beta_k^{M_k}\cdot(1-\beta_k)^{N-M_k}$$

where, $M_k=\sum_{n=1}^{N}x_k^n$

- **Prior distribution conjugate to the likelihood function**

$$p(\boldsymbol{\beta}|\boldsymbol{\alpha},L)=\prod_{k=1}^{K}p(\beta_k|\alpha_k,L)=\frac{1}{Z_1}\cdot\prod_{k=1}^{K}\beta_k^{\alpha_k}\cdot(1-\beta_k)^{L-\alpha_k}$$

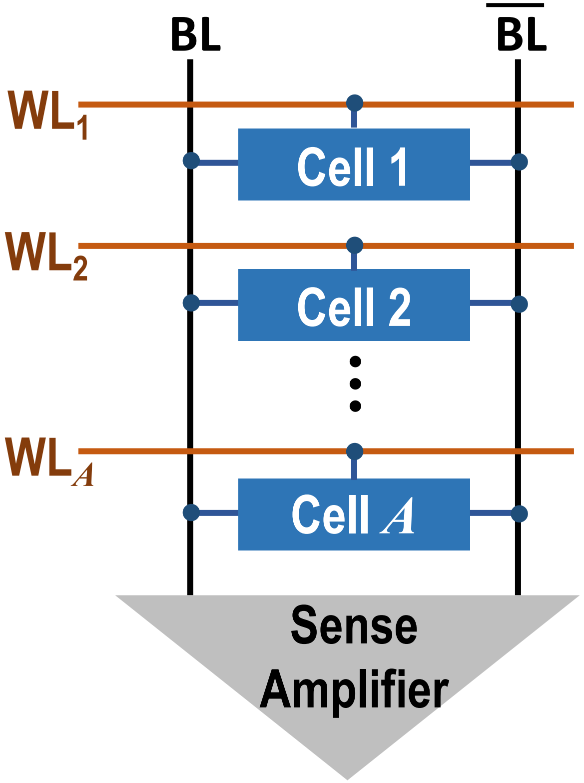
where, $Z_1=\prod_{k=1}^{K}B(\alpha_k+1,L+1-\alpha_k)$

α_k: Hyper-parameter at k-th corner
L: Hyper-parameter

The prior distribution is parametrized by the hyper-parameters $\Theta=\{L,\boldsymbol{\alpha}\}$, $\boldsymbol{\alpha}=[\alpha_1\ \alpha_2\ \cdots\ \alpha_K]^T\in\Re^K$

The priors over different corners share a unique hyper-parameter L

Bayesian Inference



Simplified SRAM array with A cells

$$\boxed{p\left(\boldsymbol{\beta} \middle| \mathrm{D}, \boldsymbol{\alpha}, L\right) \propto p\left(\boldsymbol{\beta} \middle| \boldsymbol{\alpha}, L\right) \cdot p\left(\mathrm{D} \middle| \boldsymbol{\beta}\right)}$$

- **Posterior distribution**

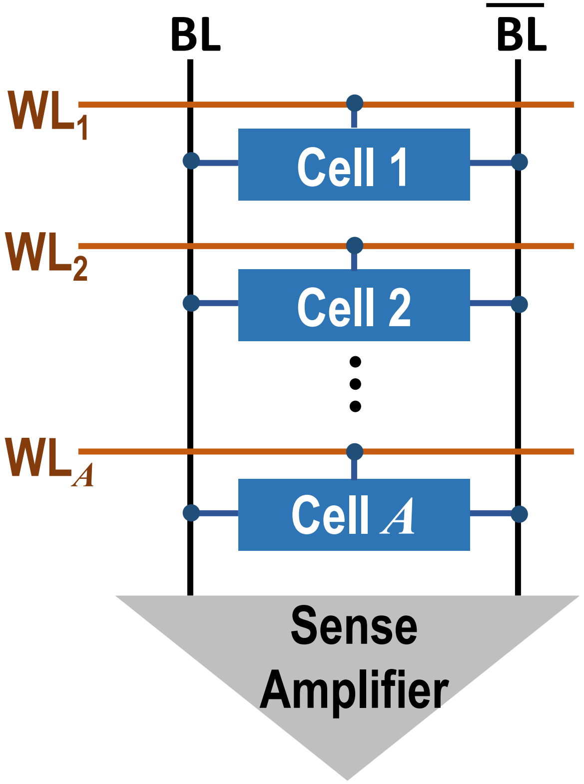
$$p\left(\boldsymbol{\beta} \middle| \mathrm{D}, \boldsymbol{\alpha}, L\right) = \frac{1}{Z_2} \cdot \prod_{k=1}^{K} \beta_k^{\alpha_k + M_k} \cdot \left(1 - \beta_k\right)^{N + L - (\alpha_k + M_k)}$$

where, $Z_2 = \prod_{k=1}^{K} B\left(\alpha_k + M_k + 1, N + L + 1 - \alpha_k - M_k\right)$

The posterior distribution is also a product of K Beta distributions

Z_2 is a normalization constant to guarantee that the integration of the posterior is equal to 1

Bayesian Inference



Simplified SRAM array with A cells

$$\boxed{p\left(\boldsymbol{\beta}\middle|\mathrm{D},\boldsymbol{\alpha},L\right)=\frac{1}{Z_2}\cdot\prod_{k=1}^{K}\beta_k^{\alpha_k+M_k}\cdot\left(1-\beta_k\right)^{N+L-(\alpha_k+M_k)}}$$

- **Maximum-A-Posteriori Estimation**

$$\ln p\left(\boldsymbol{\beta}\middle|\mathrm{D},\boldsymbol{\alpha},L\right)=\sum_{k=1}^{K}\left[\left(\alpha_k+M_k\right)\cdot\ln\beta_k+\left(N+L-\alpha_k-M_k\right)\cdot\ln\left(1-\beta_k\right)\right]-C$$

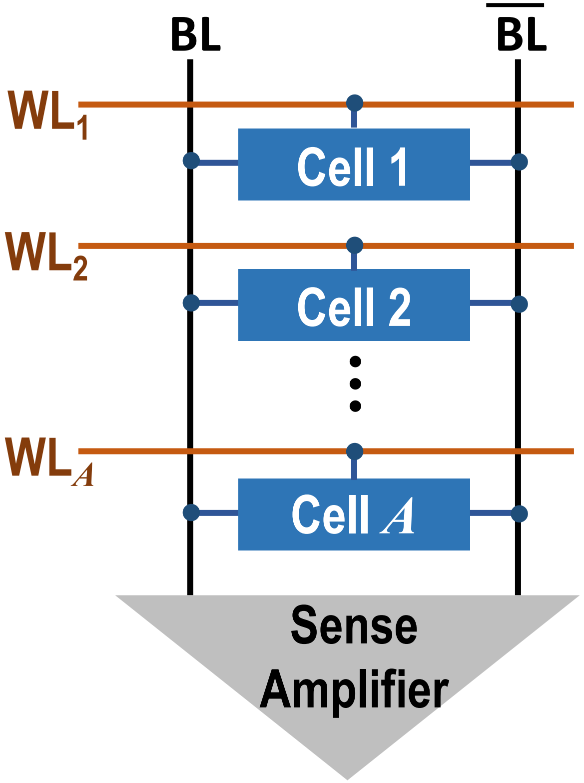
where, $C=\ln Z_2$ **Independent of β**

$$\frac{\partial}{\partial\boldsymbol{\beta}}\ln p\left(\boldsymbol{\beta}\middle|\mathrm{D},\boldsymbol{\alpha},L\right)=0$$

$$\mathbf{M}=[M_1\ M_2\ \cdots\ M_K]^T\in\Re^K$$

$$\boldsymbol{\beta}^{\mathrm{MAP}}=\frac{\boldsymbol{\alpha}+\mathbf{M}}{L+N}$$

Hyper-parameter Inference



Simplified SRAM array with A cells

$$\boxed{\Theta = \{L, \boldsymbol{\alpha}\}} \Rightarrow \max_{\boldsymbol{\alpha},L} \quad p\left(\mathrm{D}\middle|\boldsymbol{\alpha},L\right)$$

$$p\left(\mathrm{D}\middle|\boldsymbol{\alpha},L\right)=\int p\left(\mathrm{D},\boldsymbol{\beta}\middle|\boldsymbol{\alpha},L\right)d\boldsymbol{\beta}=\int p\left(\mathrm{D}\middle|\boldsymbol{\beta}\right)p\left(\boldsymbol{\beta}\middle|\boldsymbol{\alpha},L\right)d\boldsymbol{\beta}$$

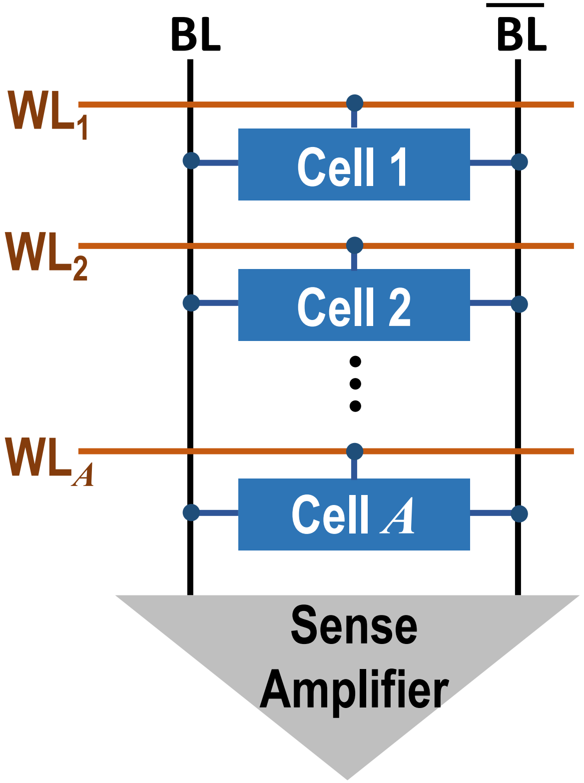
$$\Downarrow$$

$$p\left(\mathrm{D}\middle|\boldsymbol{\alpha},L\right)=\prod_{k=1}^{K}\frac{B\left(\alpha_k+M_k+1,L+N-\alpha_k-M_k+1\right)}{B\left(\alpha_k+1,L+1-\alpha_k\right)}$$

$$\Downarrow$$

$$\max_{\boldsymbol{\alpha},L} \quad \prod_{k=1}^{K}\frac{B\left(\alpha_k+M_k+1,L+N-\alpha_k-M_k+1\right)}{B\left(\alpha_k+1,L+1-\alpha_k\right)}$$

Hyper-parameter Inference



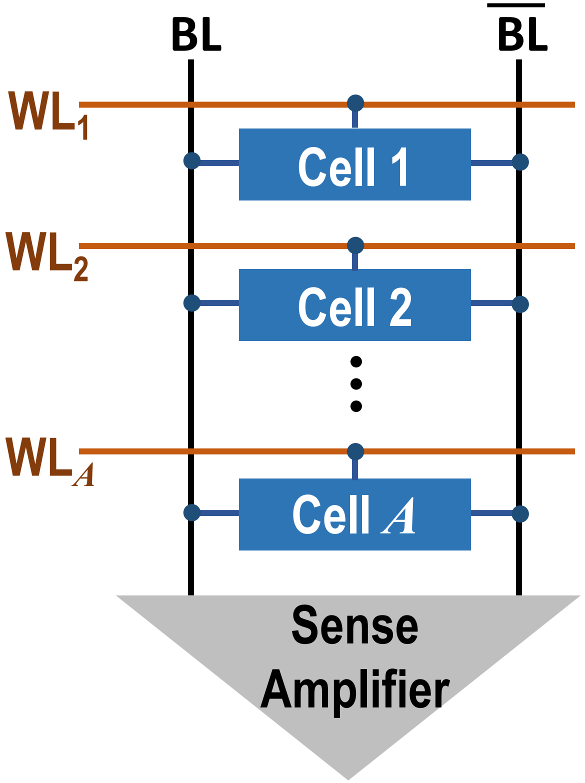
Simplified SRAM array with A cells

$$\max_{\boldsymbol{\alpha},L} \prod_{k=1}^{K} \frac{B\left(\alpha_k + M_k + 1, L + N - \alpha_k - M_k + 1\right)}{B\left(\alpha_k + 1, L + 1 - \alpha_k\right)}$$

- **Multi-start Quasi-Newton (MQN) method**
  - Hyper-parameters initialization
    - Randomly select N_s samples at each corner, where $N_s = r \times N$
    - Generate a small dataset $D_s = \{x_k^n;\ n = 1, 2, \cdots, N_s;\ k = 1, 2, \cdots, K\}$
    - $\alpha_k = M_{s,k} = \sum_{n=1}^{N_s} x_k^n \quad L = N_s$ $\quad \boldsymbol{\alpha} = [\alpha_1\ \alpha_2 \cdots \alpha_K]^T$
  - Based on hyper-parameters initialization, generate several trail points
  - Solve the optimization problem form each trail point via Quasi-Newton method
  - Choose the best solution among all local optimums as the final result.

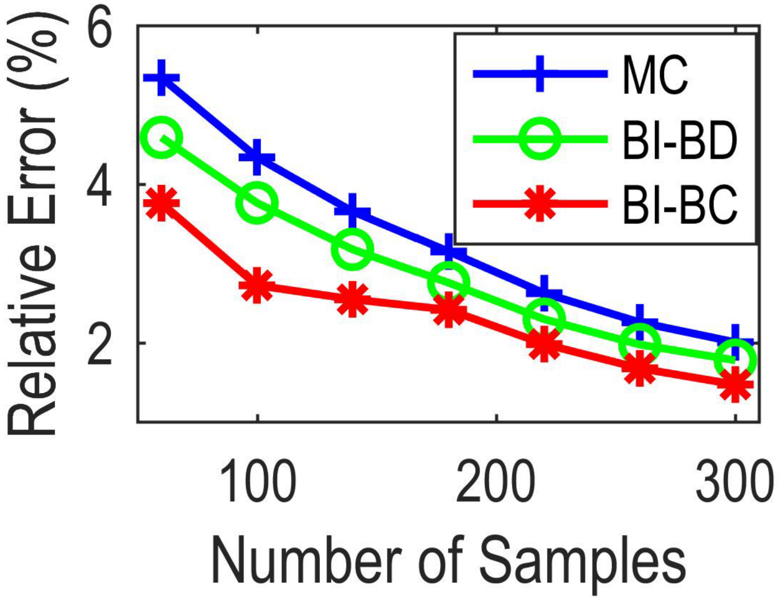
Example: SRAM Access Failure

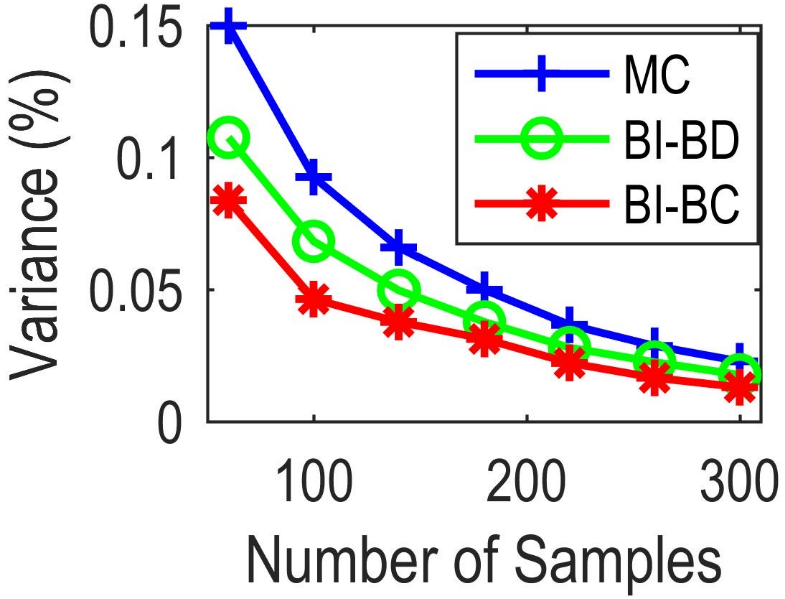
- **65nm CMOS Technology**
- **Failure definition:**
  - Differential bit-line voltage of any cell ≤ the SA input offset voltage
- **PVT corners:**
  - 5 process corners: TT, SS, FF, FS and SF



Simplified SRAM array with A cells

Example: SRAM Access Failure





	Relative Errors (%)						
# of samples	**60**	**100**	**140**	**180**	**220**	**260**	**300**
MC	5.34	4.33	3.65	3.15	2.62	2.25	2.00
BI-BD	4.59	3.76	3.17	2.75	2.29	1.97	1.76
BI-BC	3.76	2.72	2.54	2.41	1.98	1.67	1.47

Conclusion

- **Bayesian Inference method based-on Bernoulli distribution with conjugate prior (BI-BC)**
  - Using a product of Beta distributions as the conjugate prior
  - encode the performance correlation among different corners
  - Hyper-Parameter Inference by MQN method
  - BI-BC achieves up to 2.0× cost reduction over the state-of-the-art methods without surrendering any accuracy

THANKS