

# ECE 598NSG/498NSU

# Deep Learning in Hardware

# Fall 2020

## Statistical Error Compensation

Naresh Shanbhag

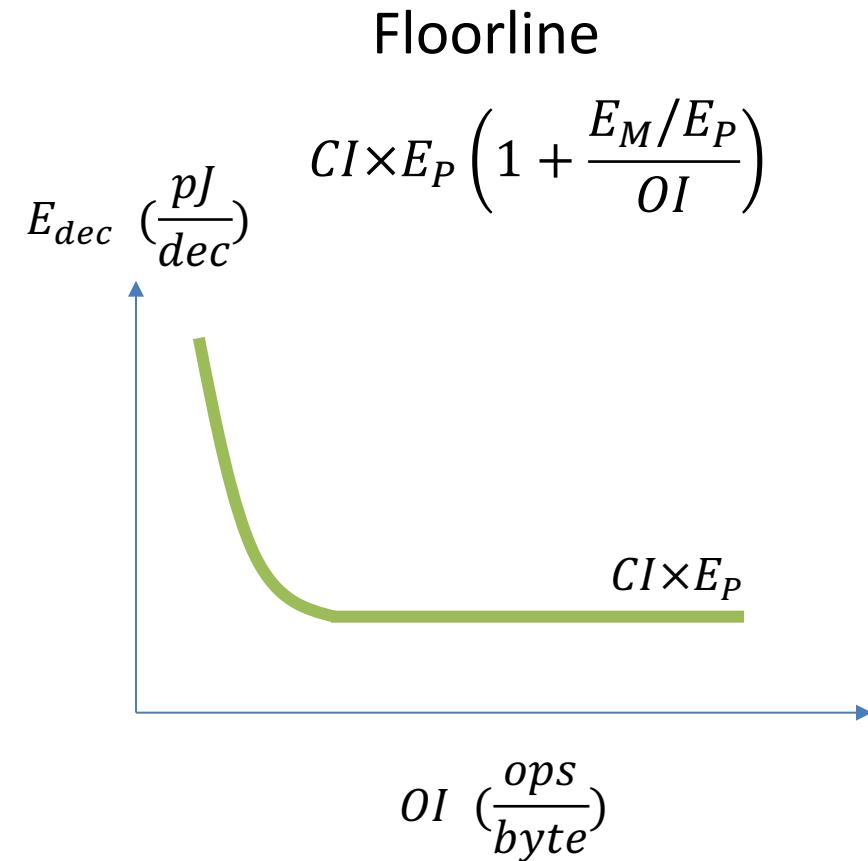
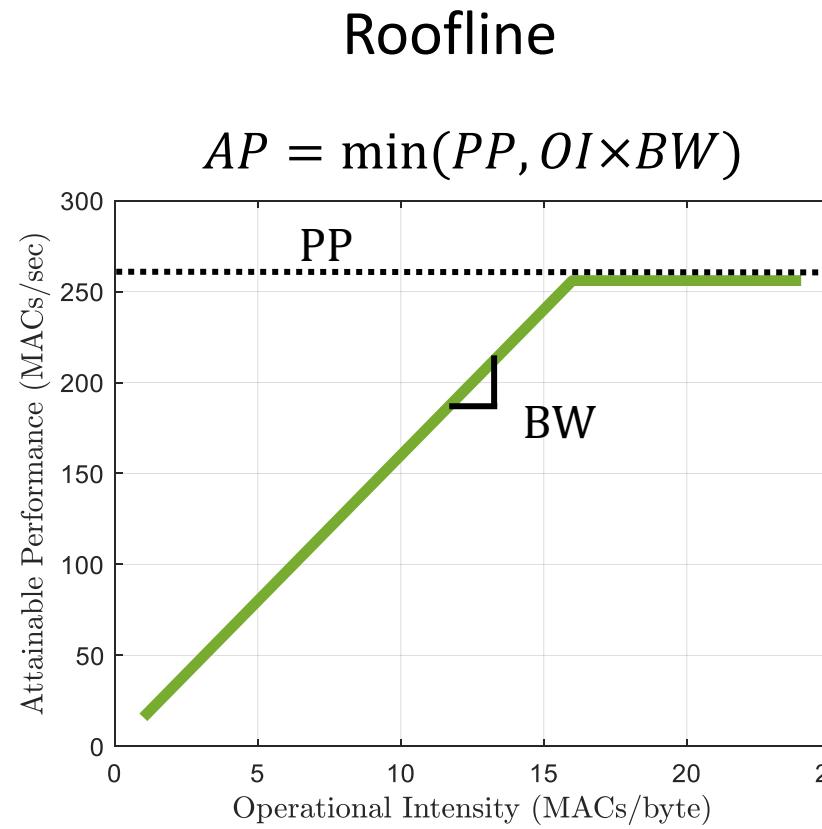
Department of Electrical and Computer Engineering  
University of Illinois at Urbana-Champaign

<http://shanbhag.ece.uiuc.edu>

# Outline

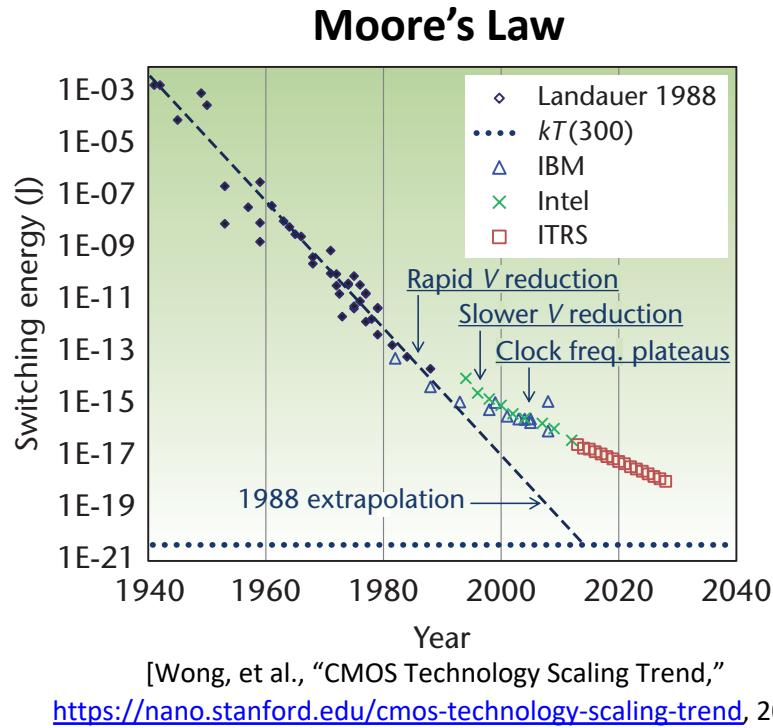
- Pushing the limits of EDP – Introduction
- Shannon-inspired Model of Computation
- Stochastic architectures/circuits
- Statistical Error Compensation (Algorithmic Noise-tolerance)
- Case Studies – Prediction-based ANT, Subthreshold ECG Classifier, Shannon-inspired Spintronics

# Reducing $\frac{E_P}{PP} = EDP$

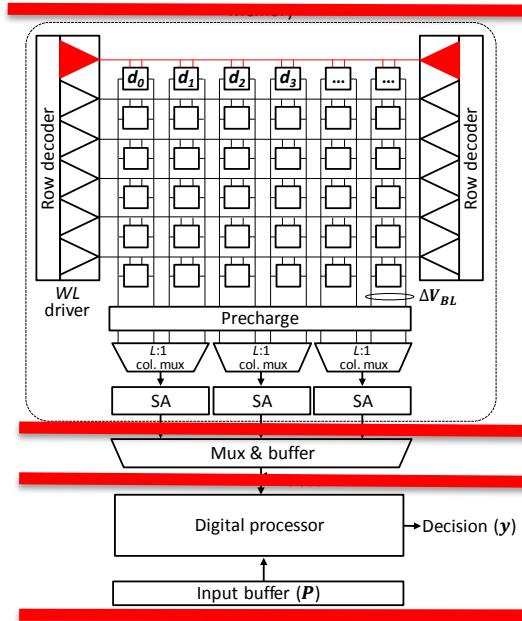


- use error compensation to minimize EDP
- can we make errors and then correct them?

# Existing Solutions are Reaching their Limits



### Von Neumann Architecture



The  
Memory Wall

- diminishing energy-delay benefits from CMOS scaling
- variations dominate → **stochasticity at the nanoscale**

- von Neumann architecture mismatched to inference workloads requirements
- data movement (**memory wall**) problem

# At the Limits of Energy-Latency-Accuracy

make errors (for efficiency) then correct (for accuracy)

Communication  
Receiver



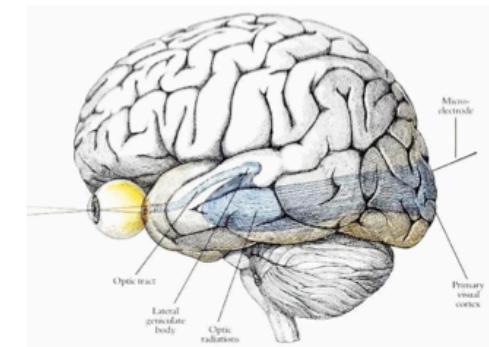
stochastic channel

STOCHASTIC COMPONENTS

+

STATISTICAL MODELS  
of  
COMPUTATION

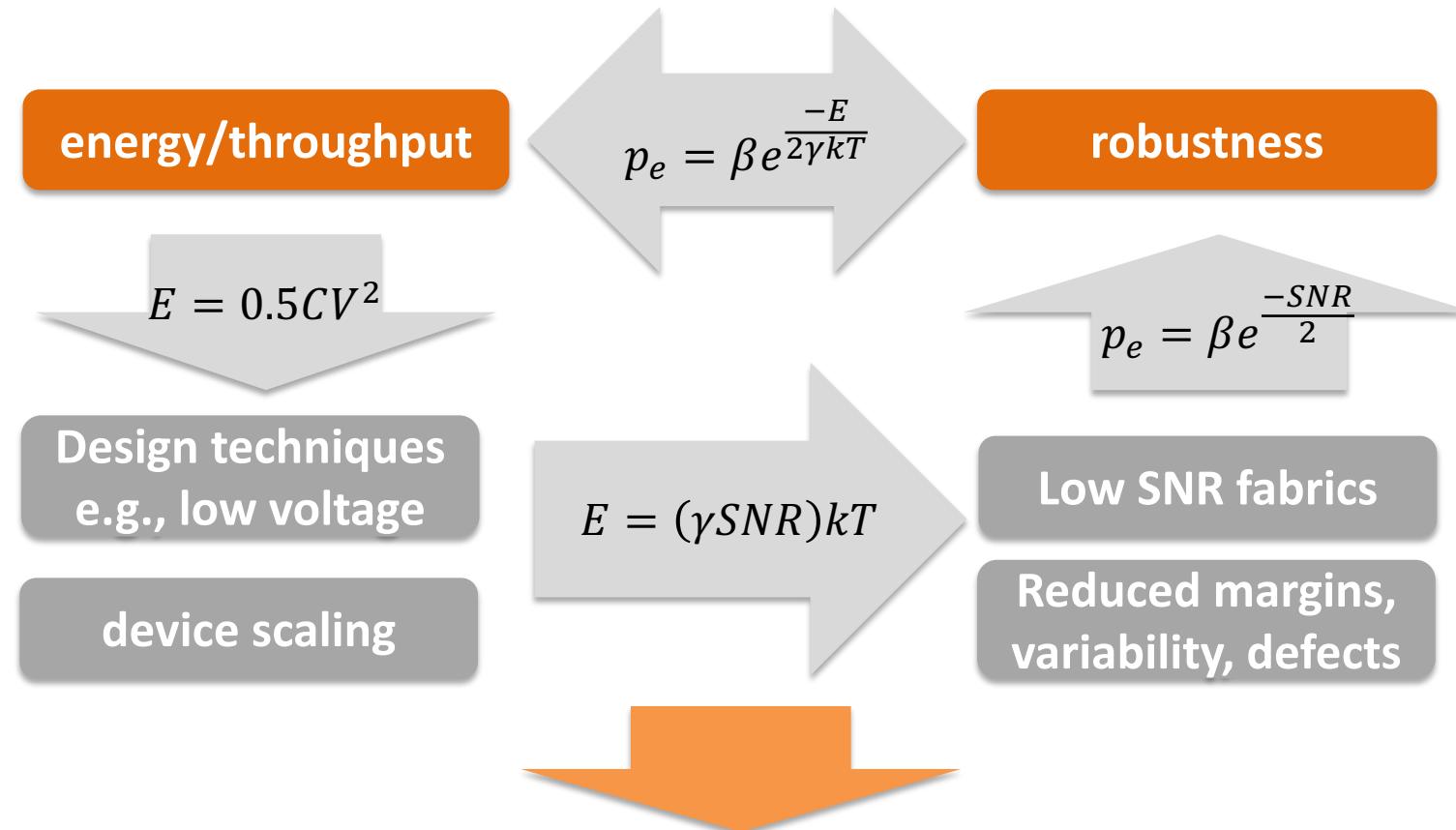
The Brain



stochastic neural fabric

How to do this in a principled manner?

# Fundamental Energy vs. Robustness Trade-off at the Limits of Scaling



Need to design **Reliable Systems** using **Unreliable Components**

# The Era of Error-Tolerant Computing

Errors will abound in future processors...and that's okay

By David Lammers

The computer's perfectionist streak is coming to an end. Speaking at the [International Symposium on Low Power electronics and Design](#), experts said power consumption concerns are driving computing toward a design philosophy in which errors are either allowed to happen and ignored, or corrected only where necessary. Probabilistic outcomes will replace the deterministic form of data processing that has prevailed for the last half century.

Naresh Shanbhag, a professor in the department of electrical and computer engineering at the University of Illinois at Urbana-Champaign, refers to error-resilient computing (also called probabilistic computing) by the more formal name of stochastic processing. Whatever the name, the approach, Shanbhag says, is not to automatically circle back and correct errors once they are identified, because that consumes power. "If the application is such that small errors can be tolerated, we let them happen," he says. "Depending on the application, we keep error rates under a threshold, using algorithmic or circuit techniques." For many applications such as [graphics processing](#) or

<https://spectrum.ieee.org/semiconductors/processors/the-era-of-errortolerant-computing>

Join IEEE | IEEE.org | [IEEE Xplore Digital Library](#) | IEEE Standards | [IEEE Spectrum](#)



Follow on: [f](#) [t](#) [in](#) [+](#) [m](#)

Engineering Topics ▾

Special Reports ▾

Blogs ▾

Advertisement



(STARnet Program by Semiconductor Research Corporation &amp; DARPA)

# Shannon-Inspired Statistical Computing for the Nanoscale Era

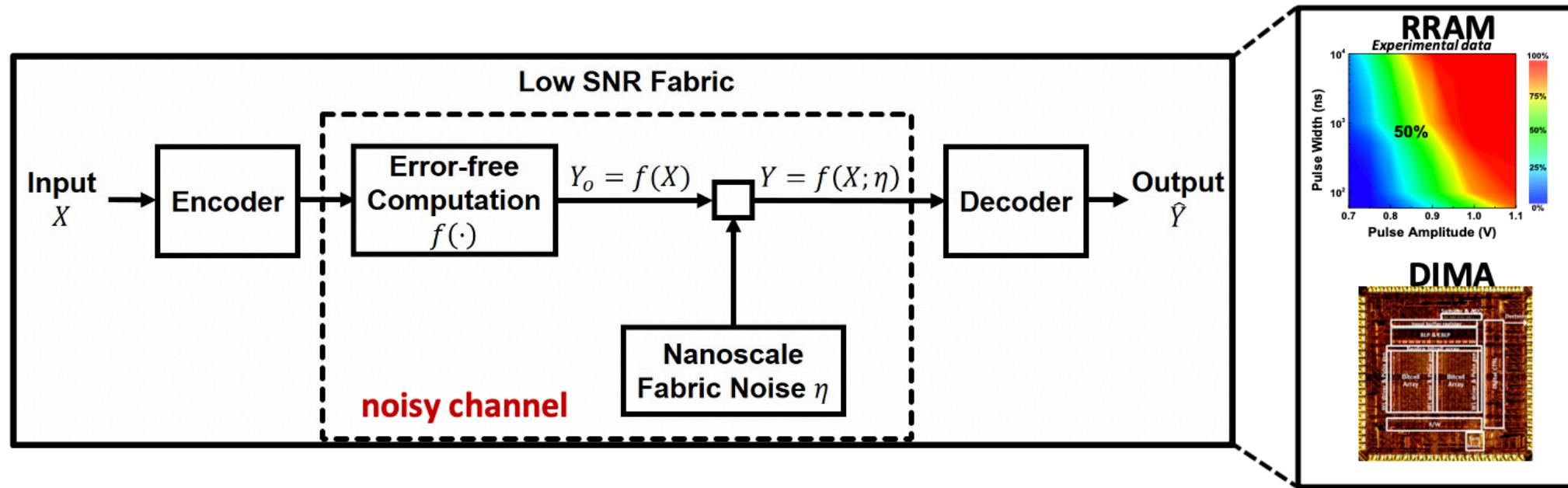
By NAresh R. SHANBHAG<sup>ID</sup>, Fellow IEEE, NAVEEN VERMA, Member IEEE,

YONGJUNE KIM<sup>ID</sup>, Member IEEE, AMEYA D. PATIL, Student Member IEEE,  
AND LAV R. VARSHNEY<sup>ID</sup>, Senior Member IEEE

Proceedings of IEEE, Special Issue on *non von Neumann Computing*, January 2019.

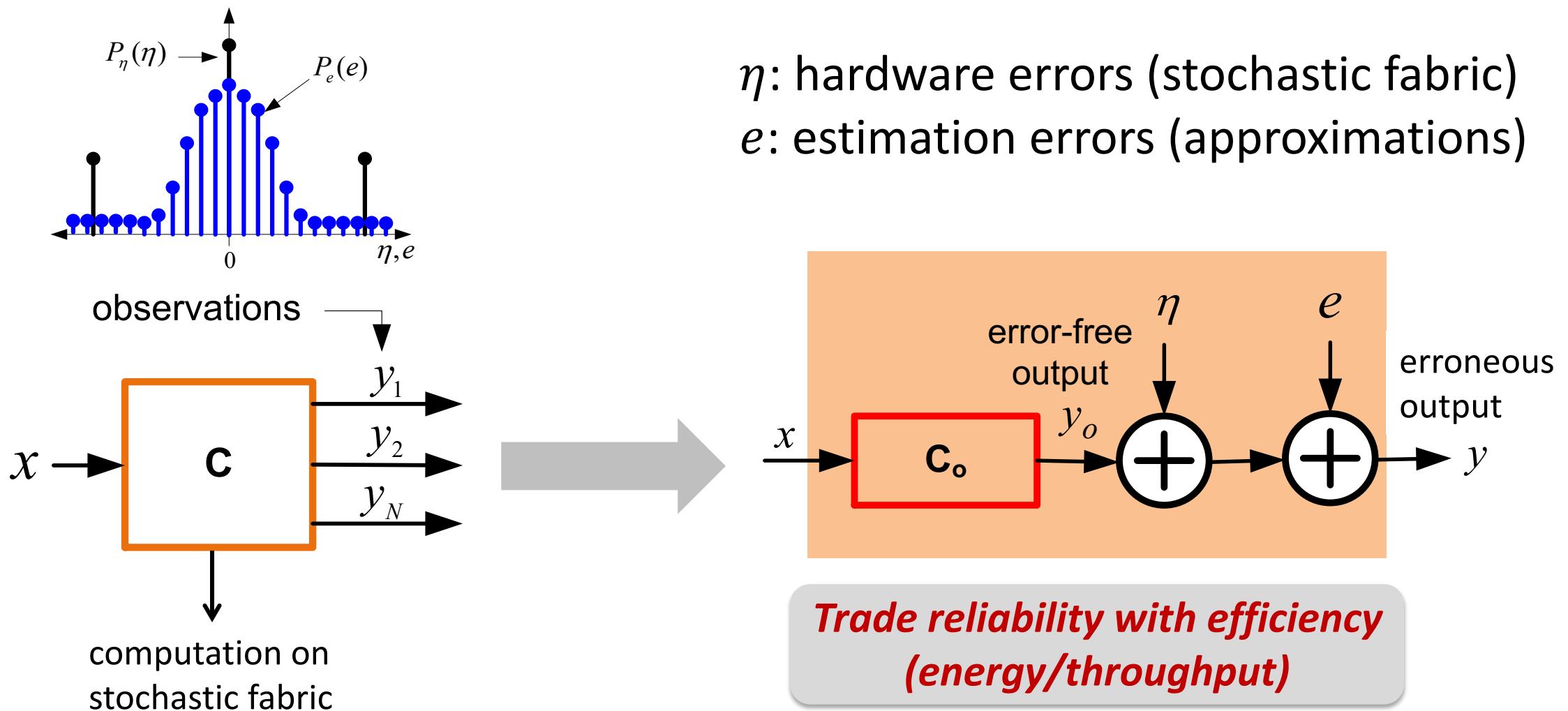
# Shannon-inspired Statistical Model of Computing

# Shannon-inspired Model of Computing

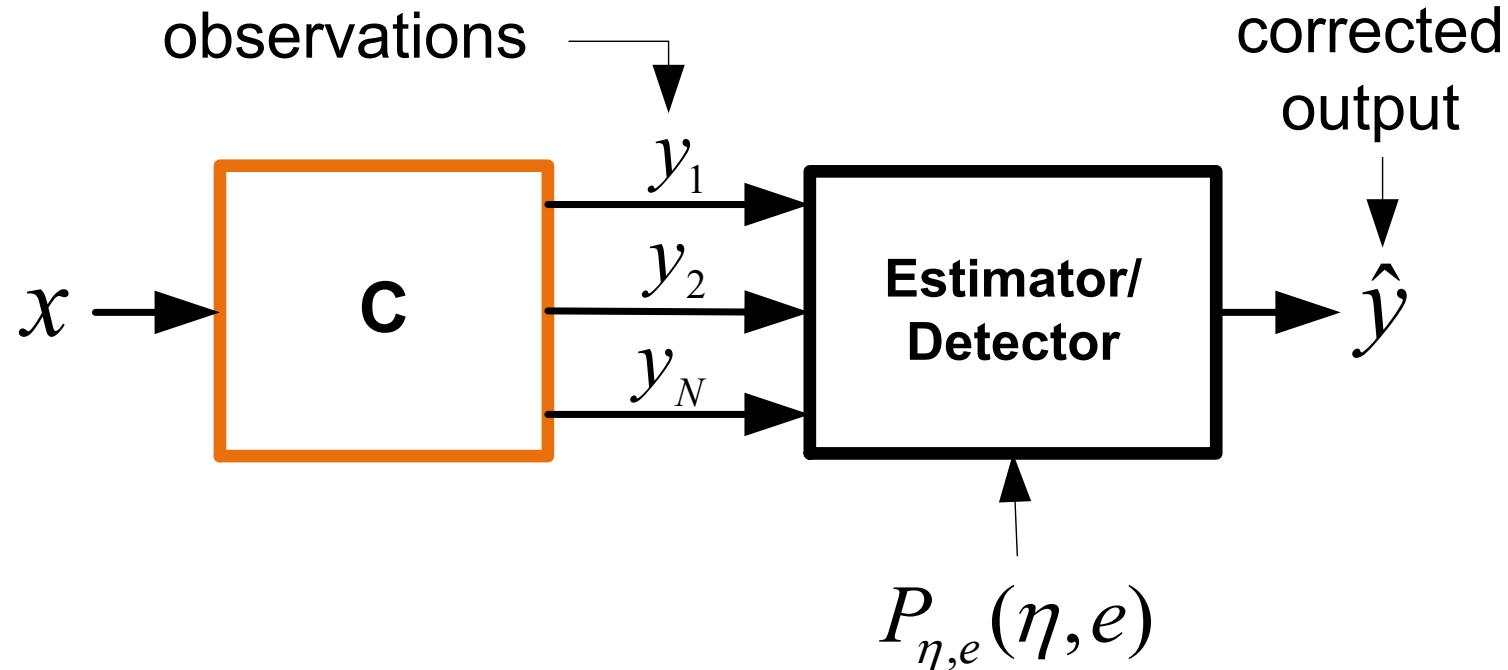


- use **information-based metrics** e.g., mutual information  $I(Y_o; \hat{Y})$
- design **low SNR fabrics** - RRAM, MRAM, deep in-memory arch. (DIMA), quantum
- develop **statistical error-compensation (SEC)** techniques

# Stochastic Nanofabric Error Model



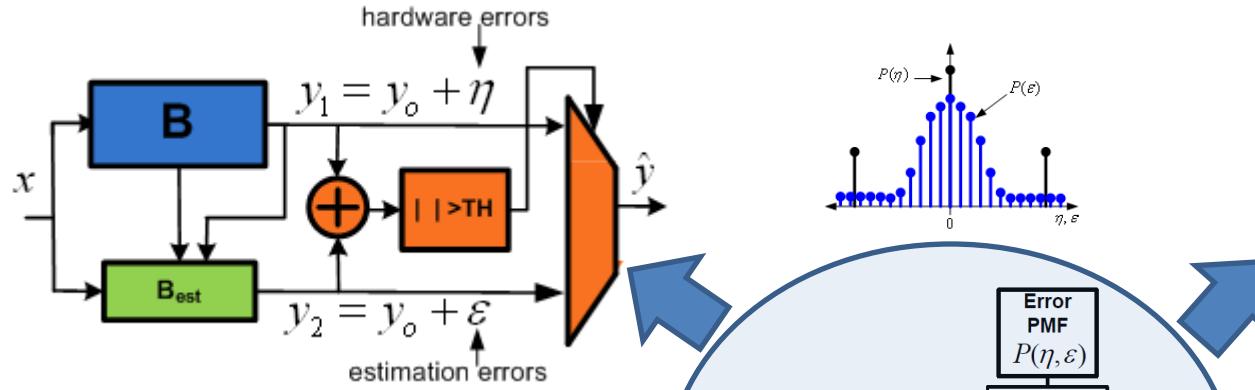
# Statistical Error Compensation (SEC)



- leverage statistical estimation & detection techniques
  - explicit or implicit use of signal and error statistics
  - machine learning extensions
- awareness of application-level metrics

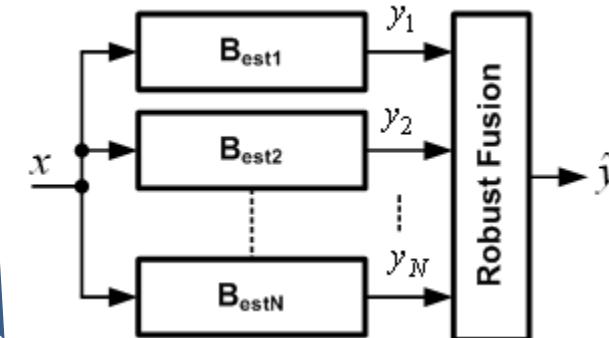
# SEC Techniques

## Algorithmic noise-tolerance (ANT)



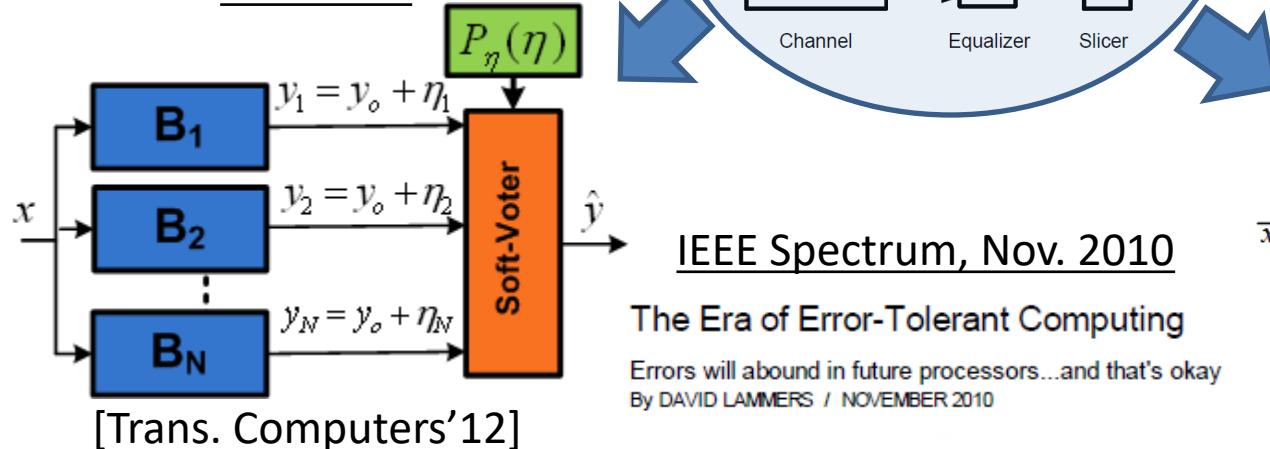
[ISLPED99,CICC01,JSSC04,  
TVLSI04,TVLSI08,JSSC13]

## Stochastic sensor NOC (SSNOC)



[TVLSI10,CICC11,TVLSI14]

## Soft NMR



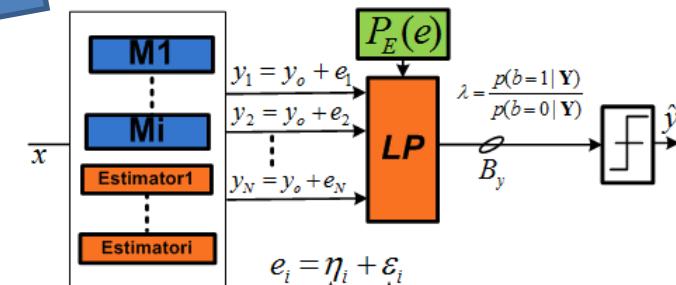
[Trans. Computers'12]

IEEE Spectrum, Nov. 2010

The Era of Error-Tolerant Computing

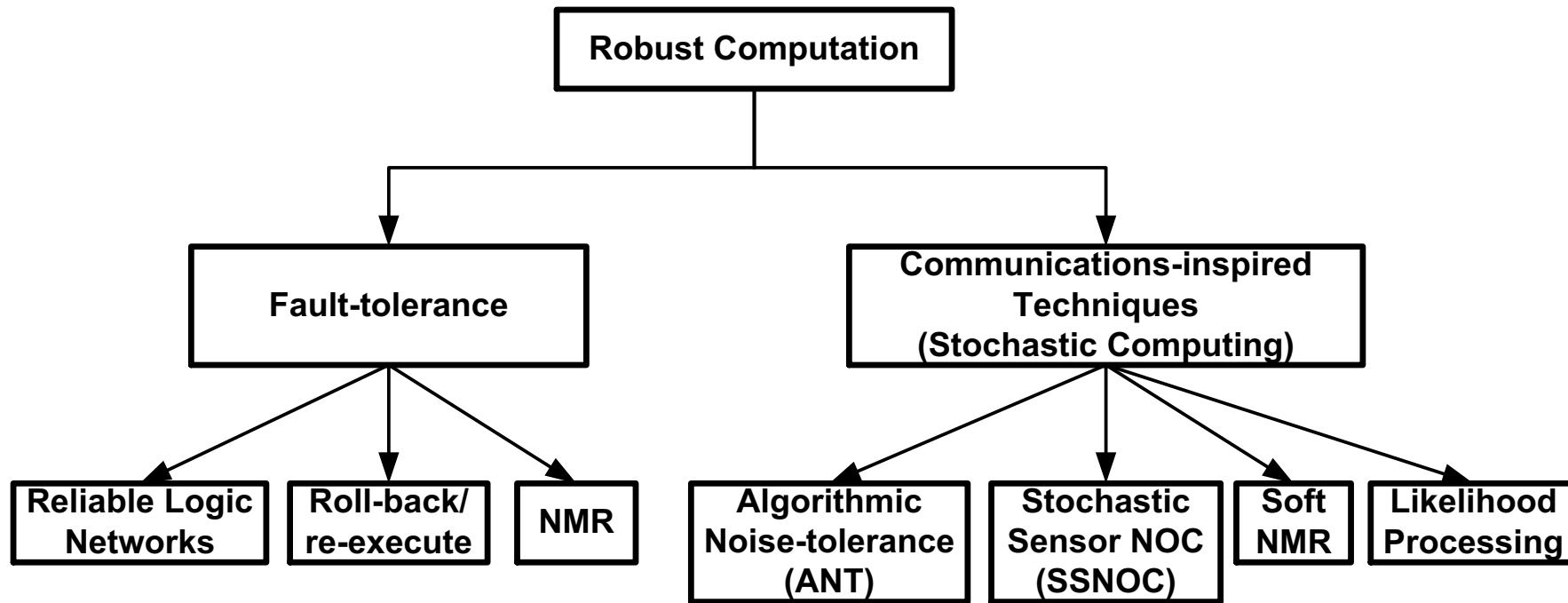
Errors will abound in future processors...and that's okay  
By DAVID LAMMERS / NOVEMBER 2010

## Likelihood Processing



[Trans. on Multimedia'13]]

# Taxonomy of Reliable Computation



- fault-tolerance → traditional approach (expensive → energy inefficient); employed in high-end servers
- Shannon/communications-inspired techniques → modern approach → addresses reliability and energy-efficiency

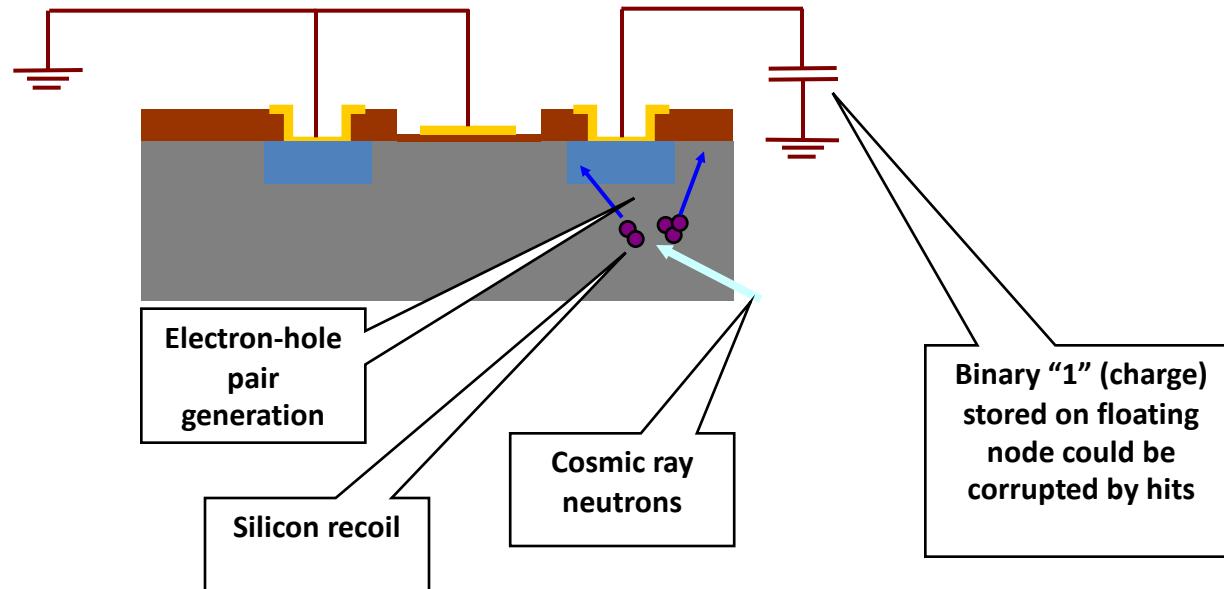
# Stochastic Architectures/Circuits

# Stochasticity in Nanoscale Circuits

- **Truly random sources:**
  - alpha particle hits; device noise (thermal, flicker, shot noise)
  - emerging nanodevices (spintronics)
- **Practically random sources:**
  - leakage noise
  - cross-talk (capacitive and inductive)
  - supply bounce
- **Aggressive design styles:**
  - dynamic low-voltage logic
  - voltage overscaling and overclocking
  - near threshold voltage (NTV) and subthreshold voltage

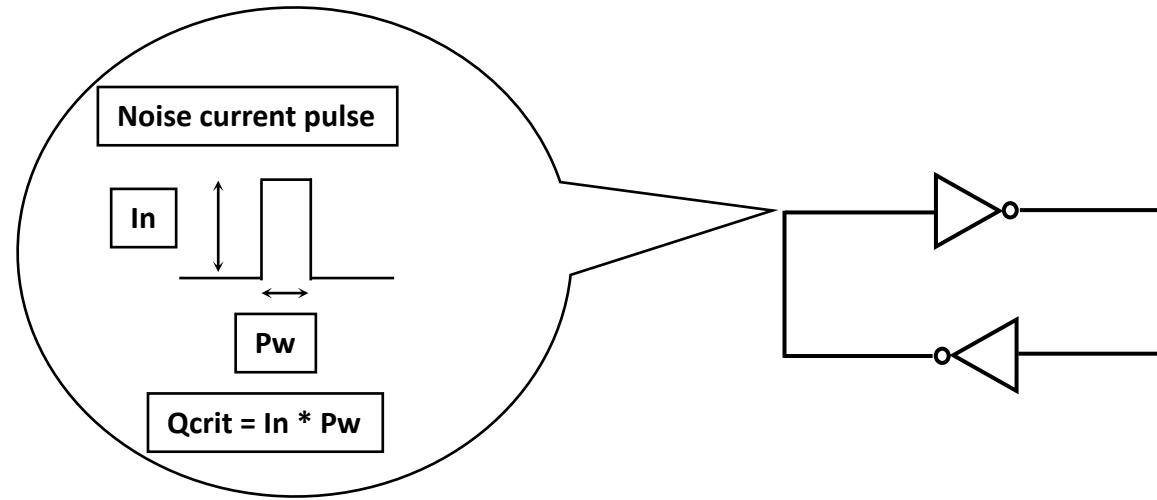
# Truly Random – Particle Hits

Categories	Sources	e-h generation
Alpha particles ( ${}^4\text{He}^2$ )	packaging material, fab material	4 to 16 fC/ $\mu\text{m}$
High-energy neutrons	cosmic rays	25 to 150 fC/ $\mu\text{m}$
Neutron-induced ${}^{10}\text{B}$ fission	interaction of cosmic ray neutrons and boron in silicon	40 fC/ $\mu\text{m}$



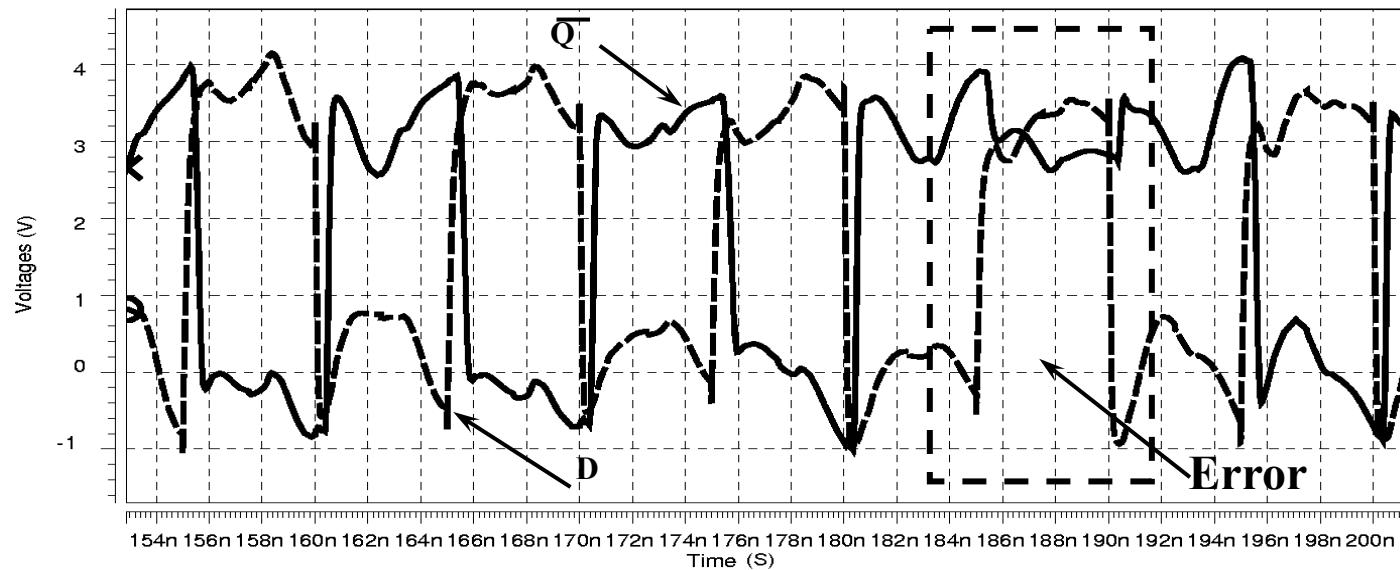
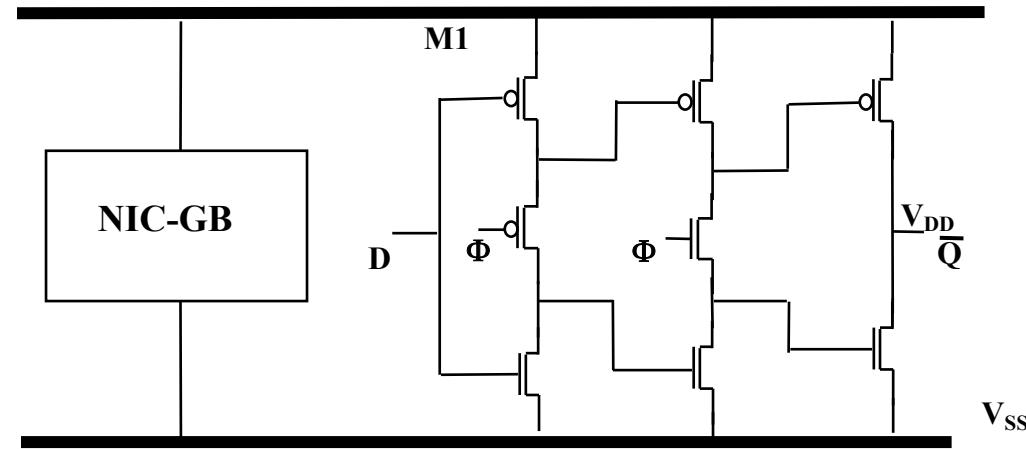
# Critical Charge

- $Q_{crit}$  : minimum amount of charge needed to upset the logic state of a circuit node

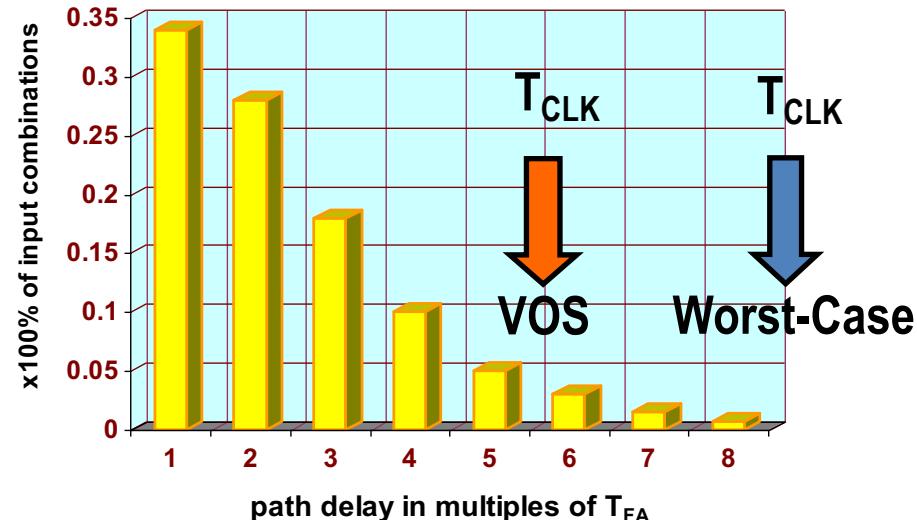


- $Q_{crit} = 11 fC$  for a minimum-sized D-Latch in  $0.18 \mu m$  CMOS
- Drain/Source junction depth  $X_J = 0.16 \mu m$
- Cosmic ray neutrons can generate charge up to  
$$Q_{gen} * XJ = 24 fC > Q_{crit}$$

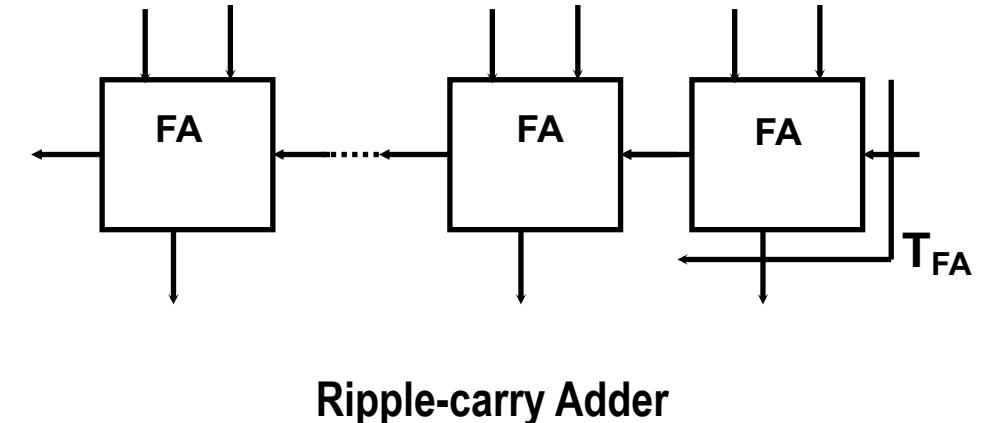
# Practically Random



# Voltage Overscaling (VOS)



Path Delay Distribution (PDD)

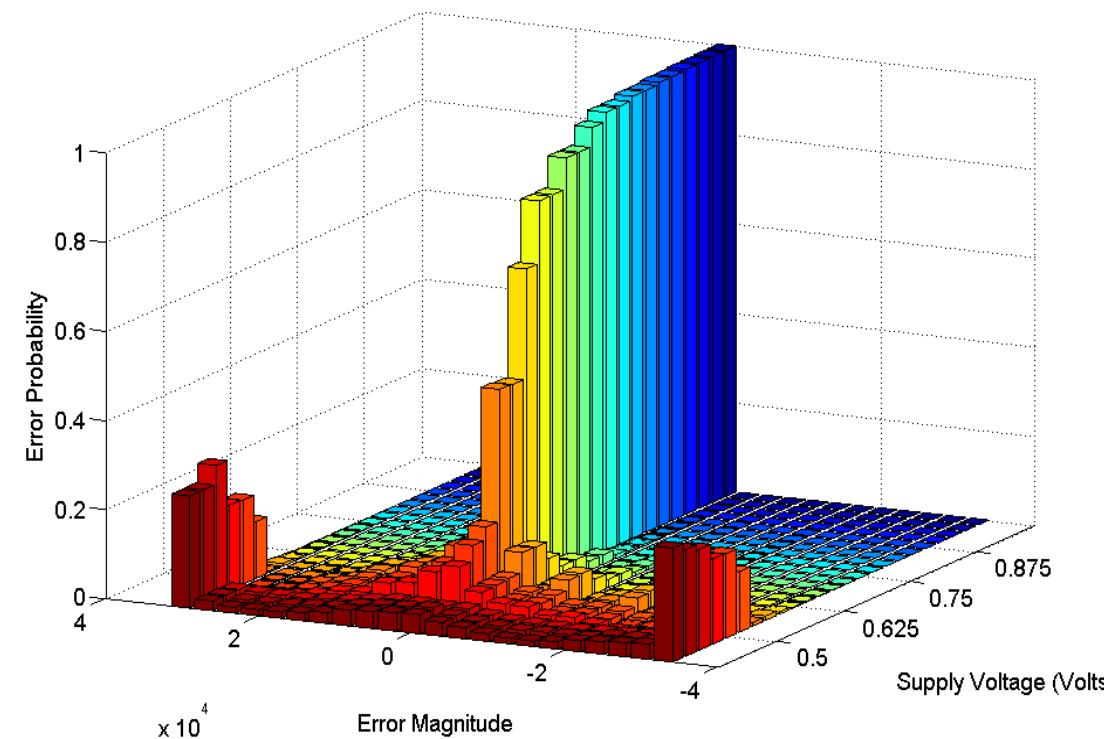


- Reduce  $V_{dd}$  such that  $T_{clk} < T_{cp}$ :

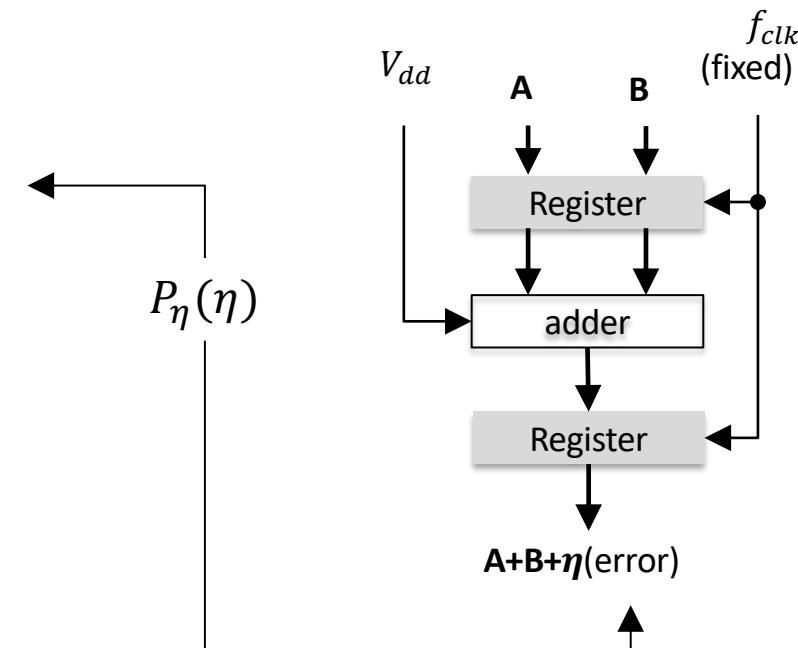
$$V_{dd} = k_{vos} V_{dd-crit}$$

$k_{vos}$ : voltage overscaling factor (VOSF);  $V_{dd-crit} = \{V_{dd} : T_{clk} = T_{cp}\}$

error probability mass function  
(16-bit ripple-carry adder)



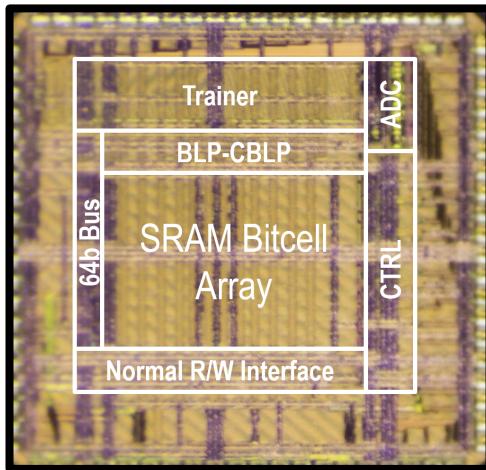
operate H/W @ voltage lower than minimum required



**9X energy savings @ error rate = 60%. How to correct errors **efficiently**?**

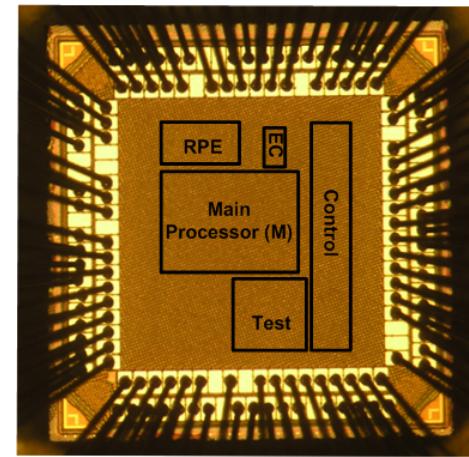
# Stochastic Hardware

## Deep In-memory Architecture



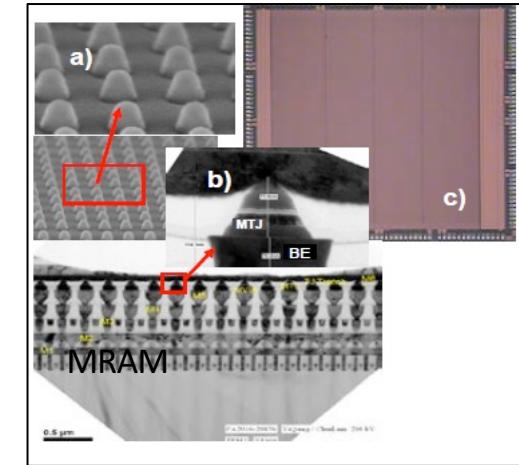
[UIUC, ISSCC'18]

## Sub/near Threshold Computing



[UIUC, JSSC'13]

## Emerging Devices



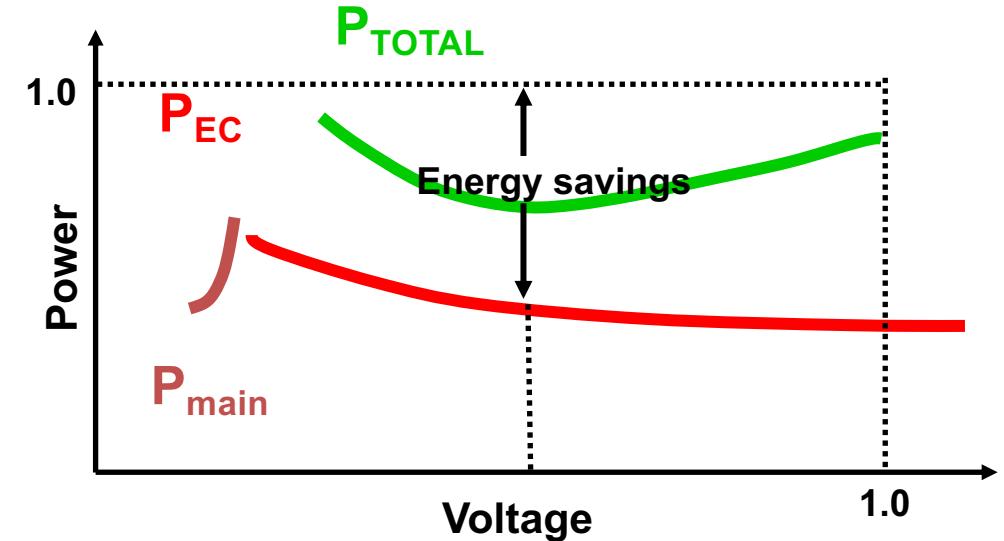
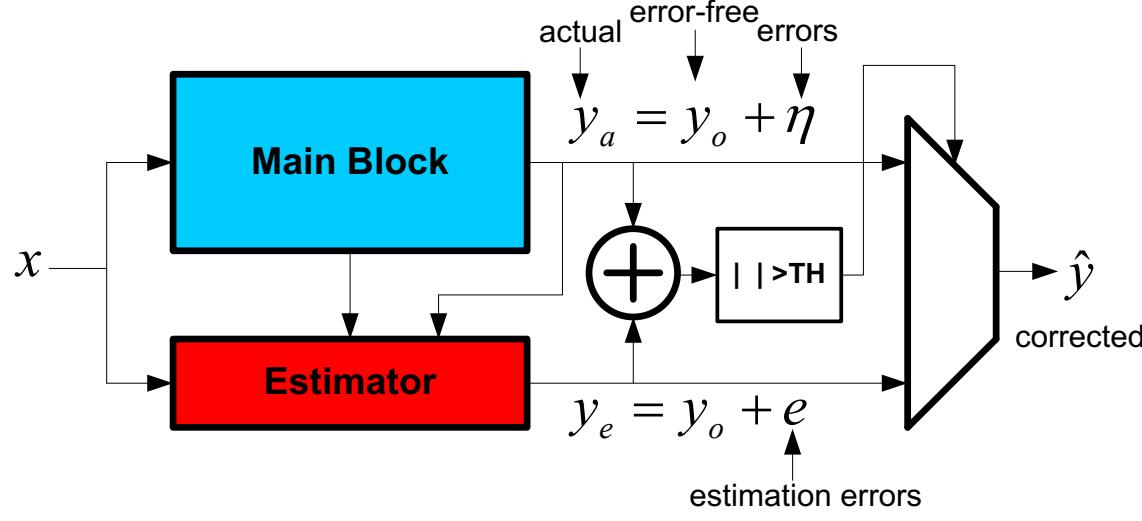
[Slaughter, IEDM'16]

- stochastic hardware  $\leftrightarrow$  low SNR fabric (The Channel)
- design fabrics with favorable SNR vs. energy trade-off

# **Algorithmic Noise-tolerance (ANT)**

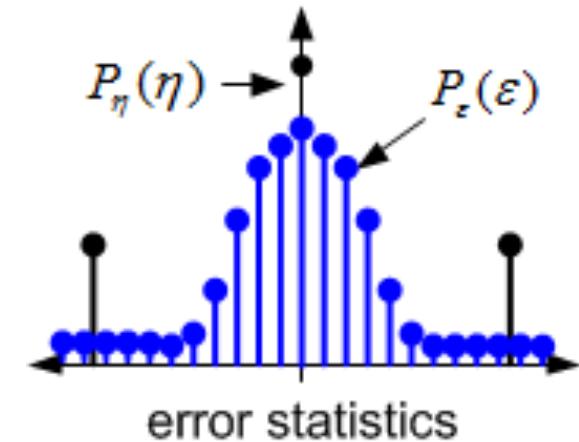
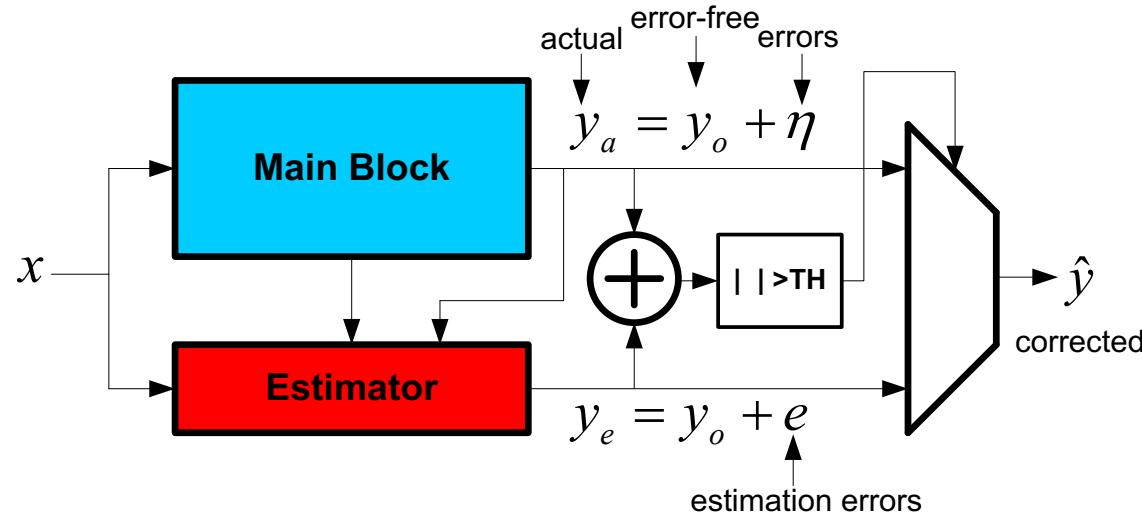
## **- An SEC Technique**

# Algorithmic Noise-Tolerance (ANT)



- high error-rates (up to 60%)
- overhead (gate-count): 5%-22%
- energy savings: 40%-70% (<1dB SNR loss)

**employ error distribution shaping** to enhance  $I(y_o; y_a, y_e)$

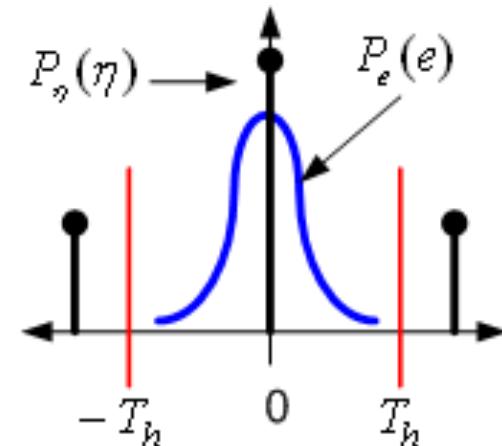
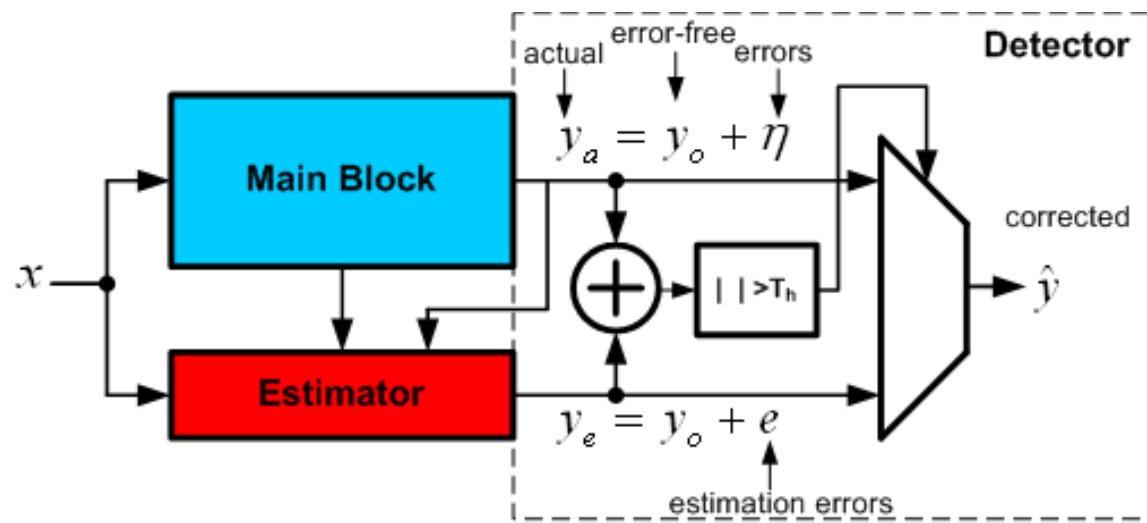


**Error detection:**  $E = 1$  if  $|y_a - y_e| > TH$  else  $E = 0$

**Error correction:**  $\hat{y} = y_e$  if  $E = 1$  else  $\hat{y} = y_a$

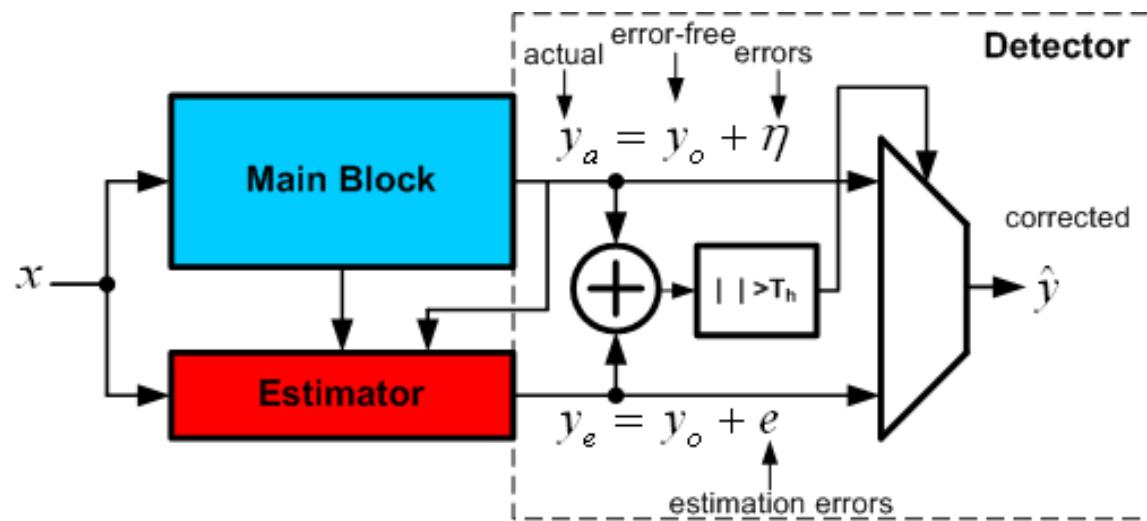
$$SNR_{main}, SNR_{est} \ll SNR_{ANT} \cong SNR_o$$

# ANT Framework



- Detector:
  - Errors → modeled as random variables/processes
- Relies on distinct H/W error and estimation error PMFs
  - H/W error PMF → spiky PMF, large magnitude
  - Estimation error PMF → e.g., Gaussian distributed
- Estimator and detector are assumed to be error-free

# Simplified SNR Analysis

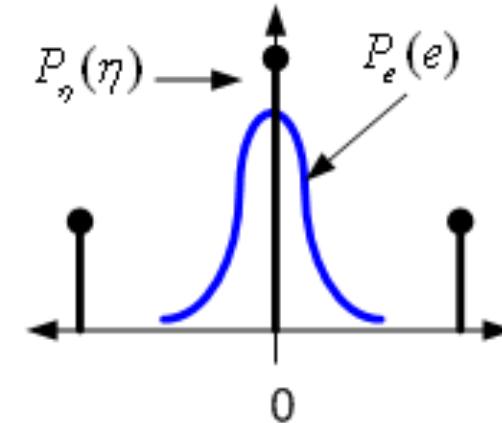


$$x = s + n; \quad y_o = y_s + n_s \rightarrow SNR_o = \frac{\sigma_{y_s}}{\sigma_{n_s}^2} \text{ (desired SNR)}$$

$$y_a = y_o + \eta \rightarrow SNR_{main} = \frac{\sigma_{y_s}^2}{\sigma_{n_s}^2 + \sigma_\eta^2} \ll SNR_o$$

$$y_e = y_o + e \rightarrow SNR_{est} = \frac{\sigma_{y_s}^2}{\sigma_{n_s}^2 + \sigma_e^2} \ll SNR_o$$

$SNR_{main} \ll SNR_{est} \ll SNR_o$  (error - free  
main - block SNR)



performance metric (SNR)

main block circuit SNR

estimator SNR

Estimator alone does not suffice

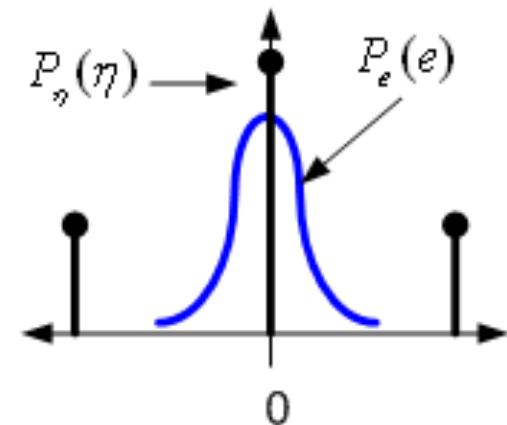
- Error compensation

$$\hat{y} = \begin{cases} y_a & \text{if } \eta = 0 \text{ (no error)} \\ y_e & \text{otherwise} \end{cases}$$

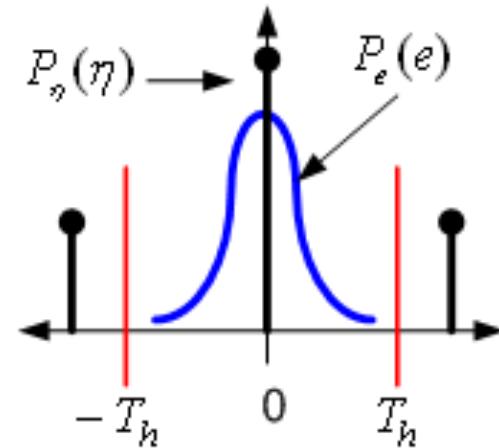
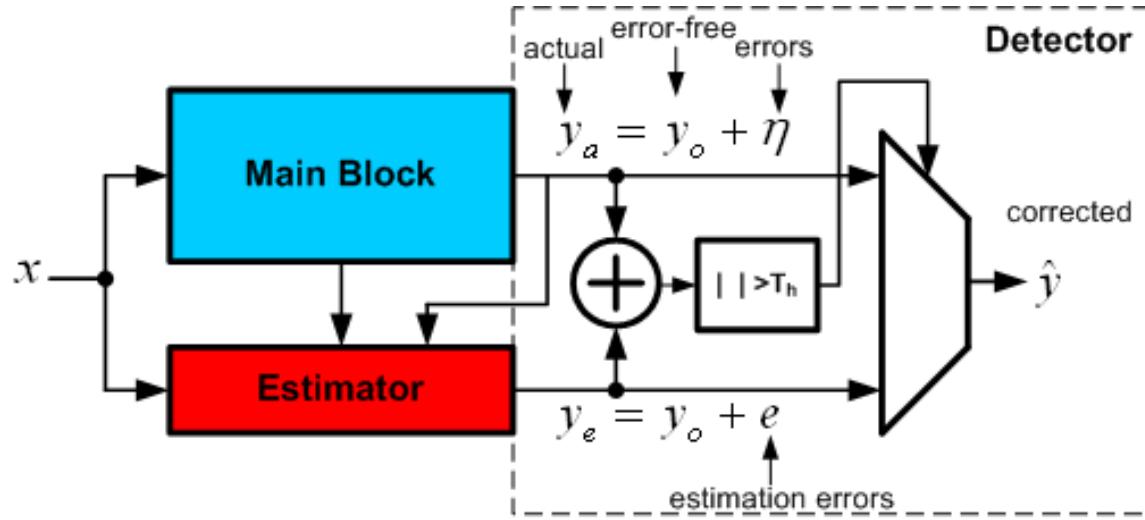
- Implemented via compare-select (simple)
- ANT system SNR similar to error-free SNR

$$SNR_{main} \ll SNR_e \ll SNR_{ANT} = \frac{\sigma_{y_s}^2}{\sigma_{n_s}^2 + \varepsilon \sigma_e^2} \approx SNR_o$$

$p_\eta = P(\eta \neq 0)$  (probability of H/W error)



# ANT Event Space



$\eta \neq 0 \rightarrow$  error event

**true -ve**

**false +ve**

**false -ve**

**true +ve**

$$P(|y_a[n] - y_e[n]| < T_h | \eta = 0) = P(|e[n]| < T_h | \eta = 0)$$

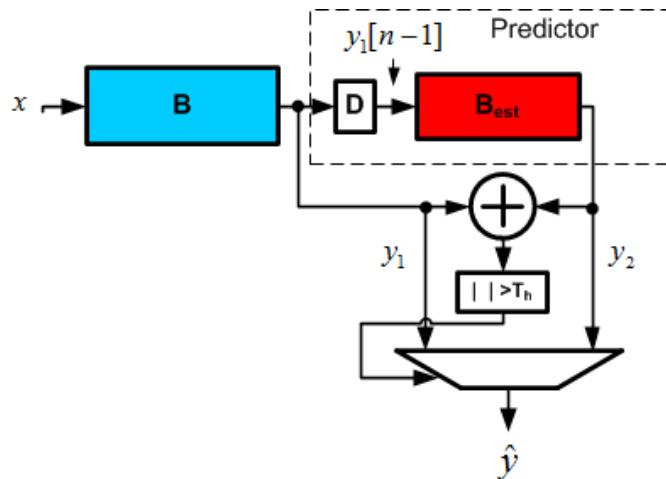
$$P(|y_a[n] - y_e[n]| > T_h | \eta = 0) = P(|e[n]| > T_h | \eta = 0)$$

$$P(|y_a[n] - y_e[n]| < T_h | \eta \neq 0) = P(|\eta[n] - e[n]| < T_h | \eta \neq 0)$$

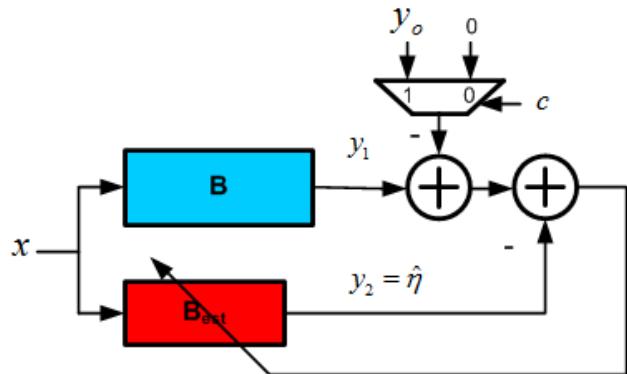
$$P(|y_a[n] - y_e[n]| > T_h | \eta \neq 0) = P(|\eta[n] - e[n]| > T_h | \eta \neq 0)$$

# ANT Techniques

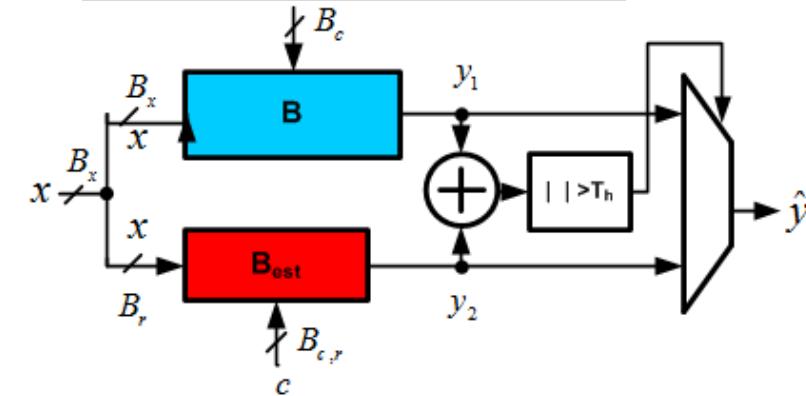
**prediction-based**



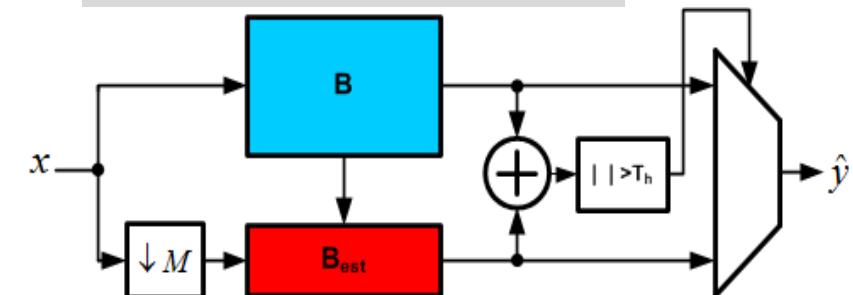
**adaptive error-cancellation**



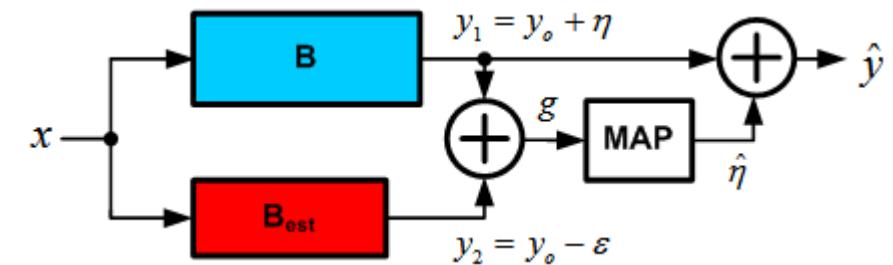
**reduced-precision replica**



**input subsampled replica**



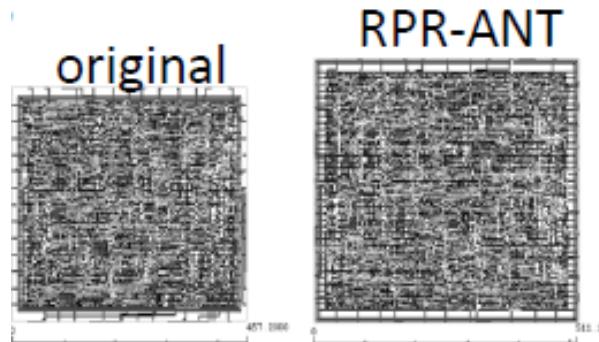
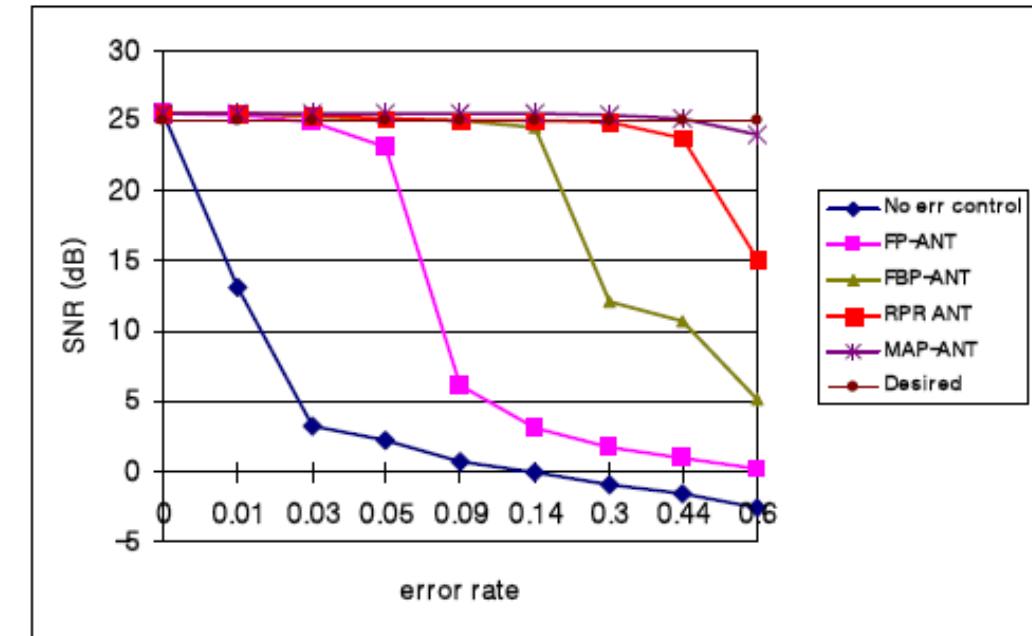
**maximum a posteriori (MAP)**



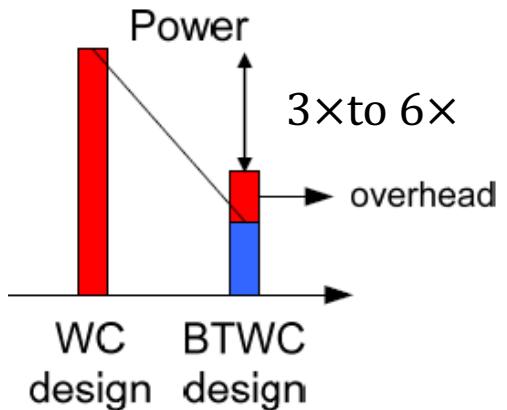
# ANT Overhead

## SNR Performance

Block	Taps	Mult	Add	Gate – Count	Complexity Overhead%
Main Block	32	16x16	33	34944	0
FP ANT	4	8X8	17	1184	3.4
FPB ANT	5	8X8	17	1480	4.2
RPR ANT	28	8X8	17	7696	22
MAP ANT	28	8X8	17	7800	22.3



high error-rates (up to 60%)  
complexity overhead: 5%-22%  
energy savings: 3×to 6×  
latency overhead:  $t_{\text{adder}} + t_{\text{comp}} + t_{\text{mux}}$

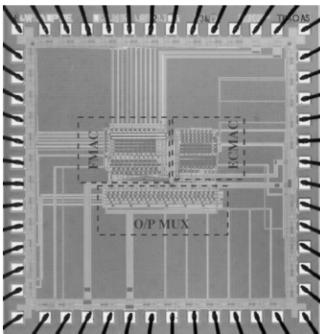


# Shannon-inspired IC Prototypes

**5X-to-6X (energy efficiency) over critical voltage operation @ iso-accuracy and iso-latency**

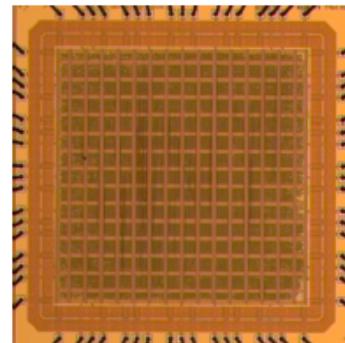
**90%-to-95% accuracy @ raw error rates of 58%-to-86%**

**Bandpass FIR Filter**  
(350nm)



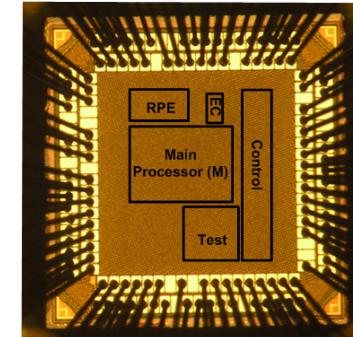
[Hedge, JSSC 2004]

**CDMA PN code Acquisition**  
(180nm)



[Kim, CICC 2011]

**Subthreshold ECG Classifier**  
(45nm)



[Abdallah, JSSC 2013]

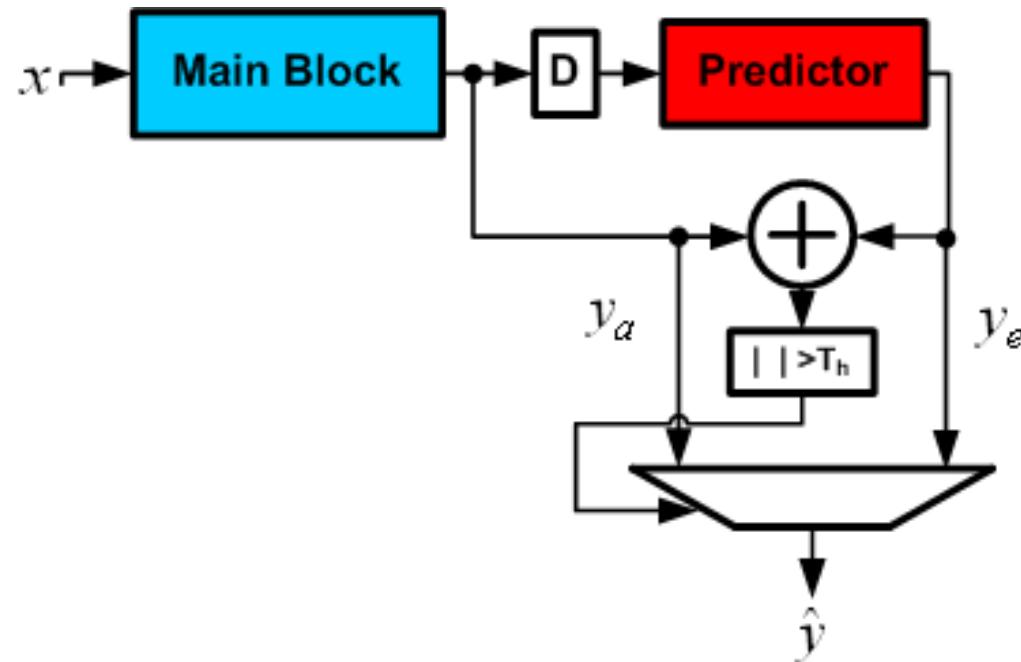
# Case Study I – Prediction-based ANT Prototype IC

A Voltage Overscaled Low-Power Digital Filter IC

Rajamohana Hegde and Naresh R. Shanbhag

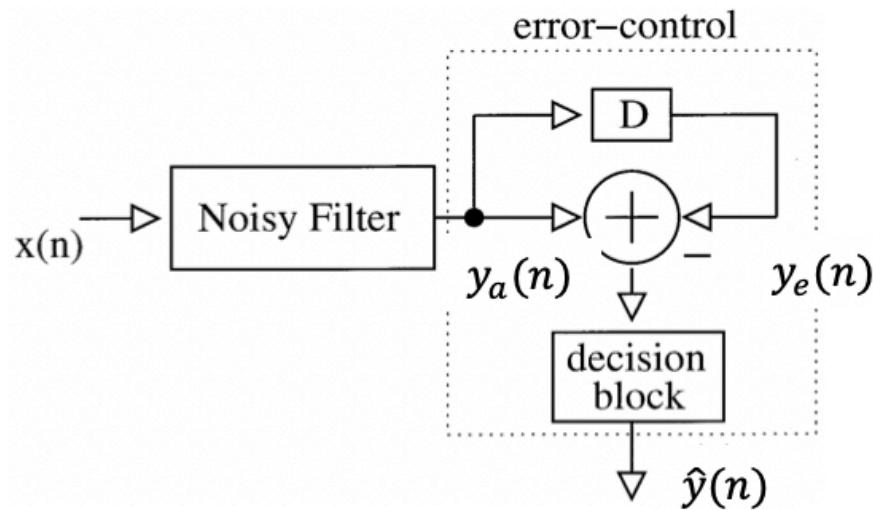
IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 39, NO. 2, FEBRUARY 2004

# Prediction-based ANT



- assume main block output is highly correlated
- use a predictor of  $y_a$  to generate  $y_e$
- compute optimal (Wiener-Hopf) predictor coefficients off-line or adaptively

# Difference-based Prediction



$$\begin{aligned}y_e(n) &= y_a(n - 1) \\y_a(n) &= y_o(n) + \eta(n) \\e(n) &= y_a(n) - y_e(n) \\&= y_o(n) - y_o(n - 1) + \eta(n)\end{aligned}$$

- If no error in  $y_a(n - 1)$
- $|y_o(n) - y_o(n - 1)|$  small for correlated (narrowband) outputs

- **Error compensation:**

- Compute  $e(n)$
- If  $|e(n)| \geq T_h$  : declare error and corrected output as  $\hat{y}(n) = y_a(n - 1)$ ;  
Otherwise:  $\hat{y}(n) = y_a(n)$
- Overhead: 2 adders, 1 latch

# General Prediction-based ANT

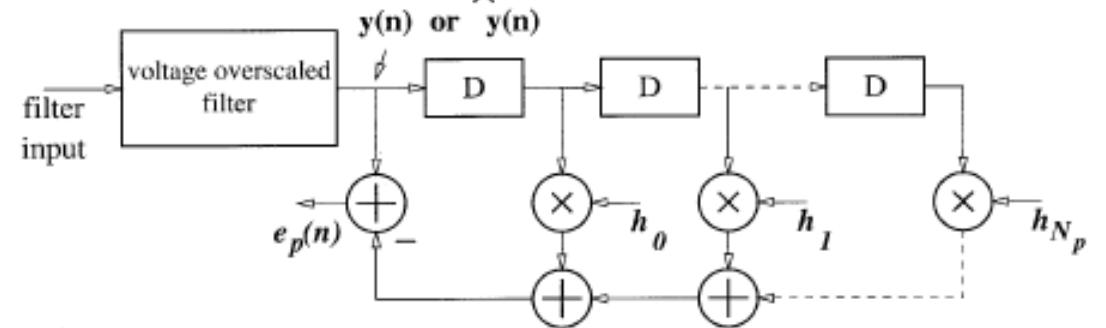
$$y_o[n] = \sum_{k=0}^{N-1} h_k x[n-k]$$

$$y_a[n] = y_o[n] + \eta[n]$$

$$y_e[n] = \sum_{k=0}^{N_p-1} p_k y_a[n-k-1]$$

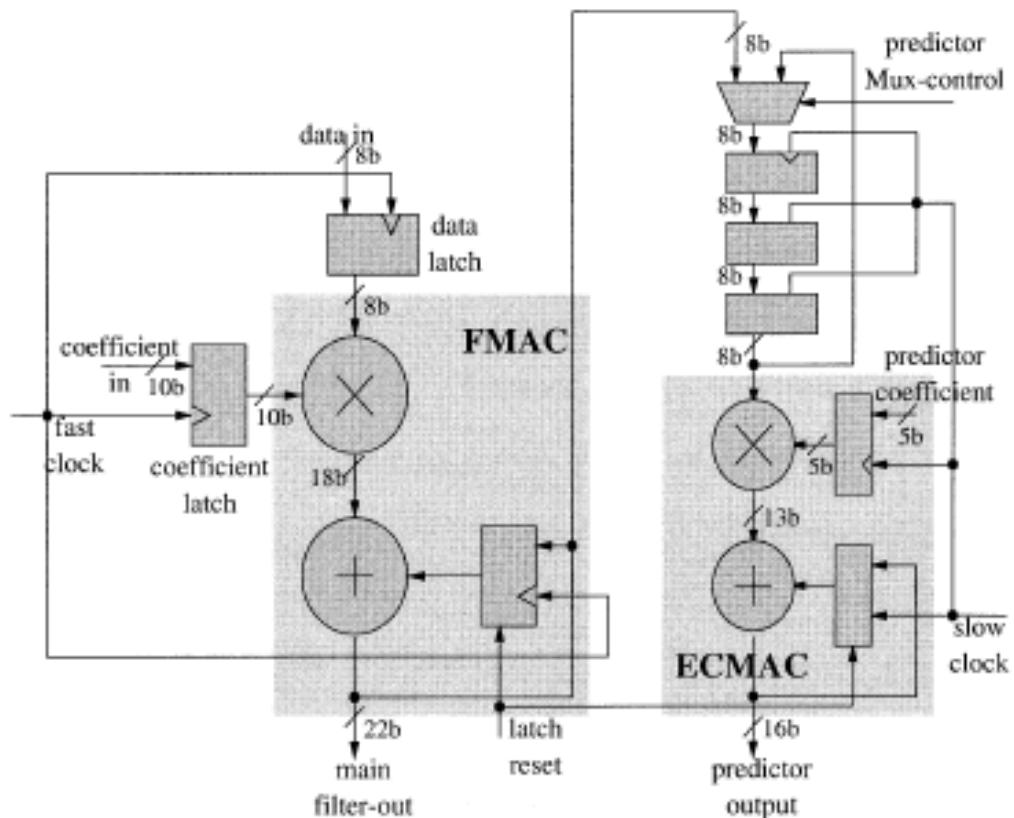
$$y_a[n] - y_e[n] = y_o[n] + \eta[n] - y_e[n]$$

$$= e[n] + \eta[n] \longrightarrow \text{Assuming 1 error in } N_p+1 \text{ samples}$$



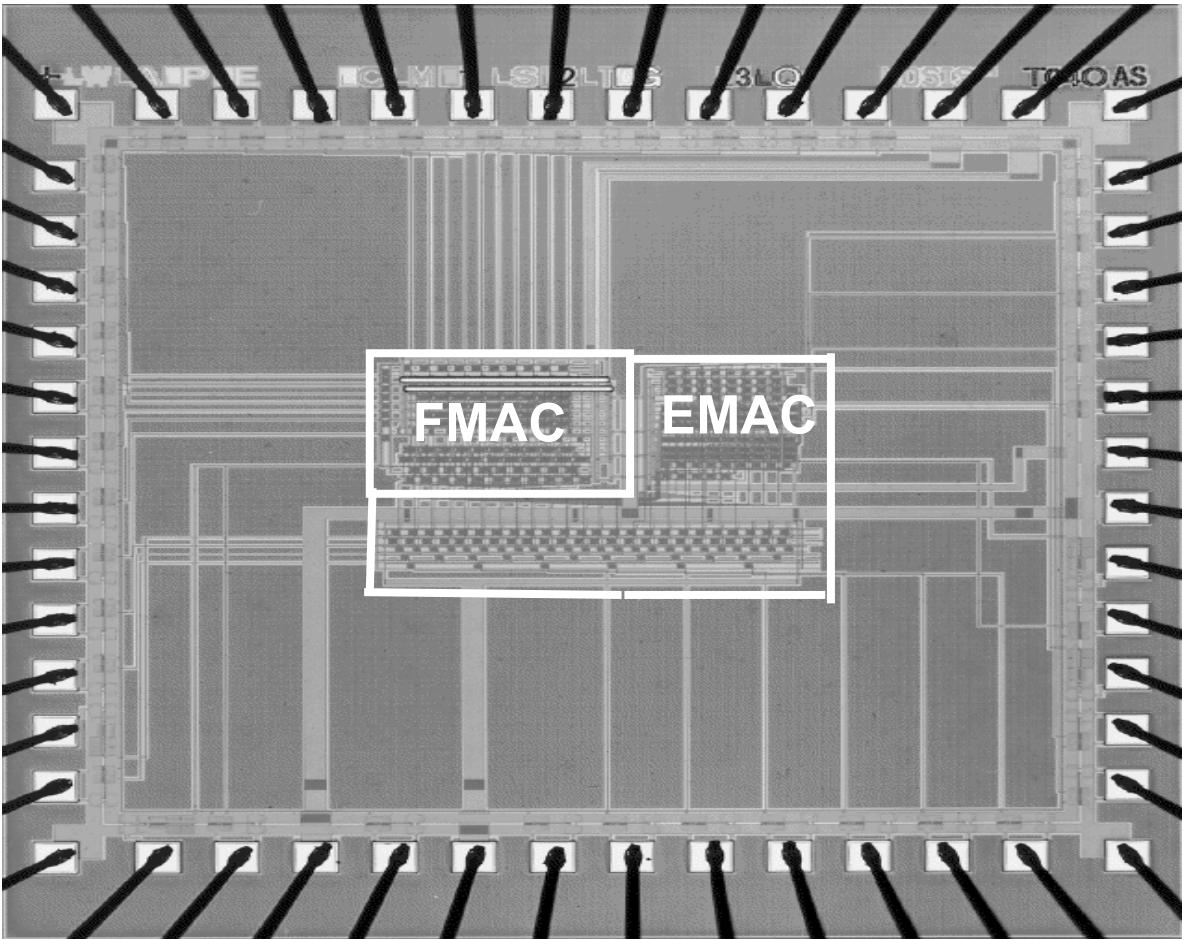
- Can employ backward, forward-backward and non-linear prediction

# IC Architecture



- folded architecture
- FMAC: 10-b  $\times$  8-b signed multiplier + 22-b adder
- Predictor (EMAC) – 8X **slower clock** than FMAC  $\rightarrow$  32-tap main filter + 4-tap predictor
- EMAC has **smaller precision** – 5-b  $\times$  8-b multiplier + 16-b adder
- EMAC will fail much later than FMAC

# Die Photo



Hegde and Shanbhag [JSSC04]

## FMAC:

- Clock: 88MHz
- $V_{dd-crit}$ : 3.55V
- Power at critical supply: 105.47 mW (whole chip)
- Lowest supply: 2.32V
- 3872 transistors

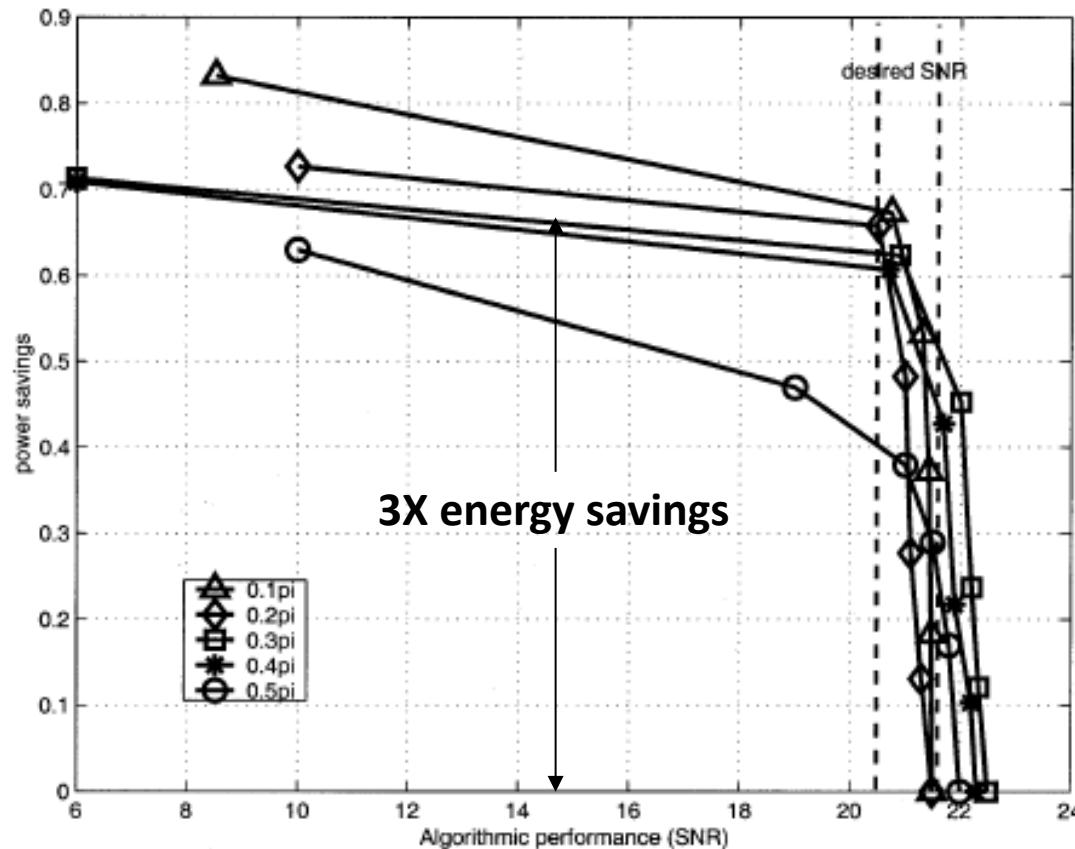
## EMAC:

- Clock: 11MHz
- $V_{dd-crit}$ : 2.25V
- 2423 transistors

# Simulation and Measured Results

$$PS = \frac{P_{\text{ref}} - P_{\text{vos}}}{P_{\text{ref}}}$$

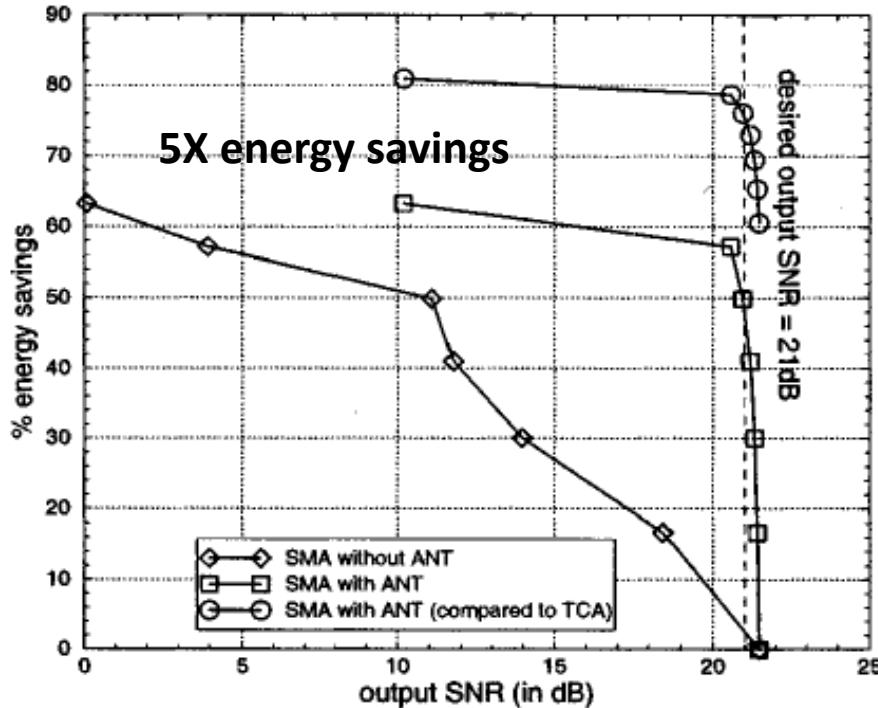
40%-67% power savings for BW:  $0.05f_s - 0.25f_s$   
with  $< 1\text{dB}$  loss in SNR (loss in accuracy)



0.35 $\mu\text{m}$ ; 3.3V CMOS  
32-tap FIR; 4-tap pred

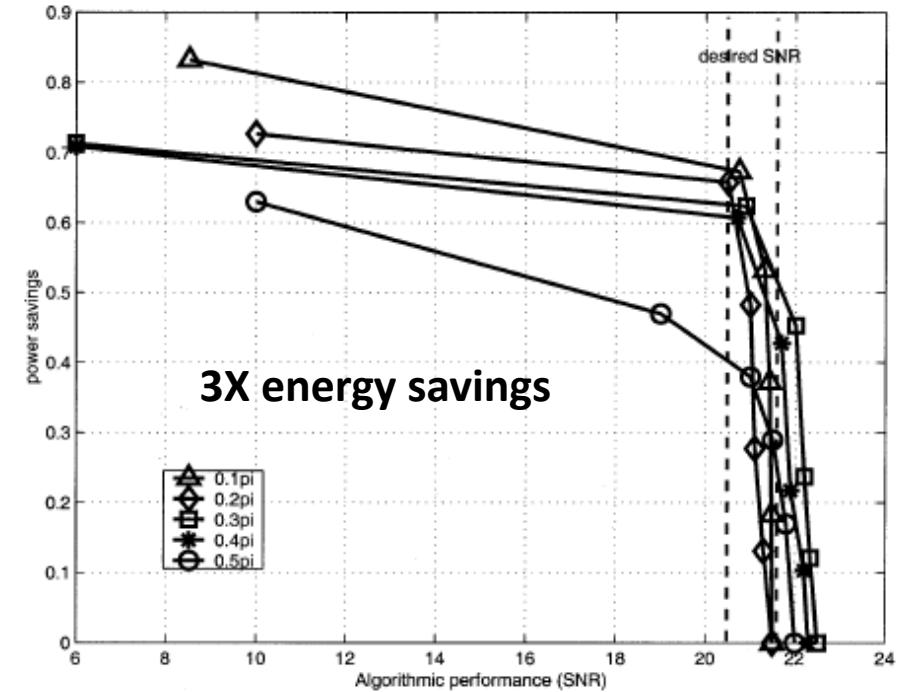
# Matching Simulation and Measured Results

## simulation results



3-tap predictor;  $0.5\pi$  BW  
29-tap FIR

## measured results

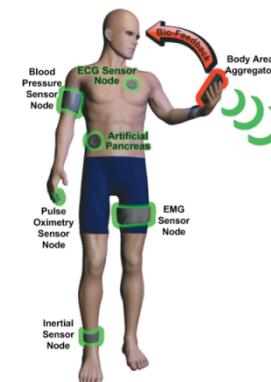


$0.35\mu\text{m}$ ; 3.3V CMOS  
32-tap FIR; 4-tap predictor

# Case Study - II

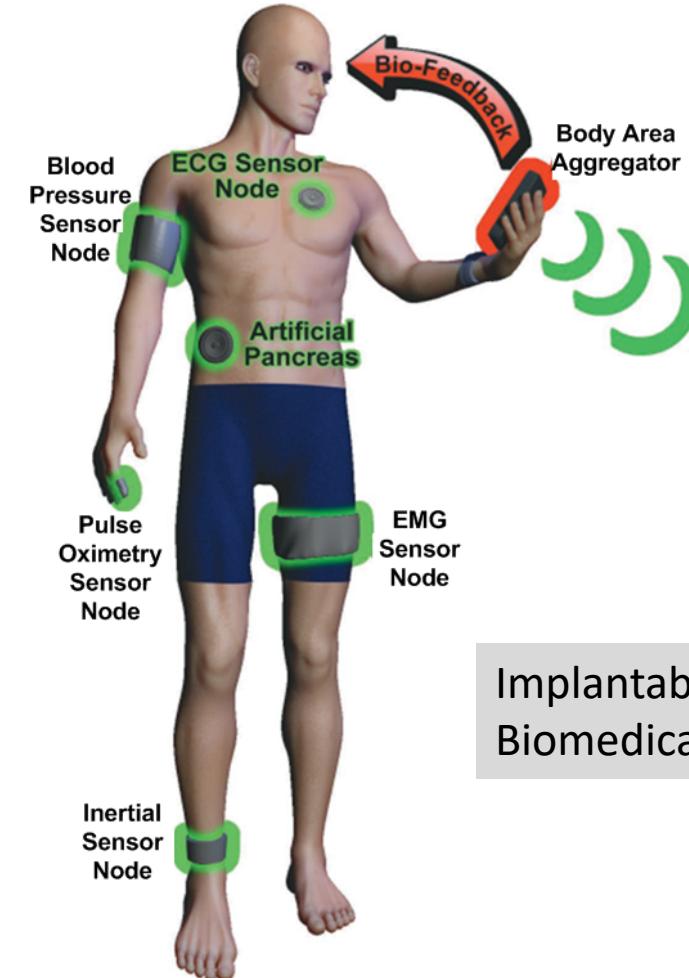
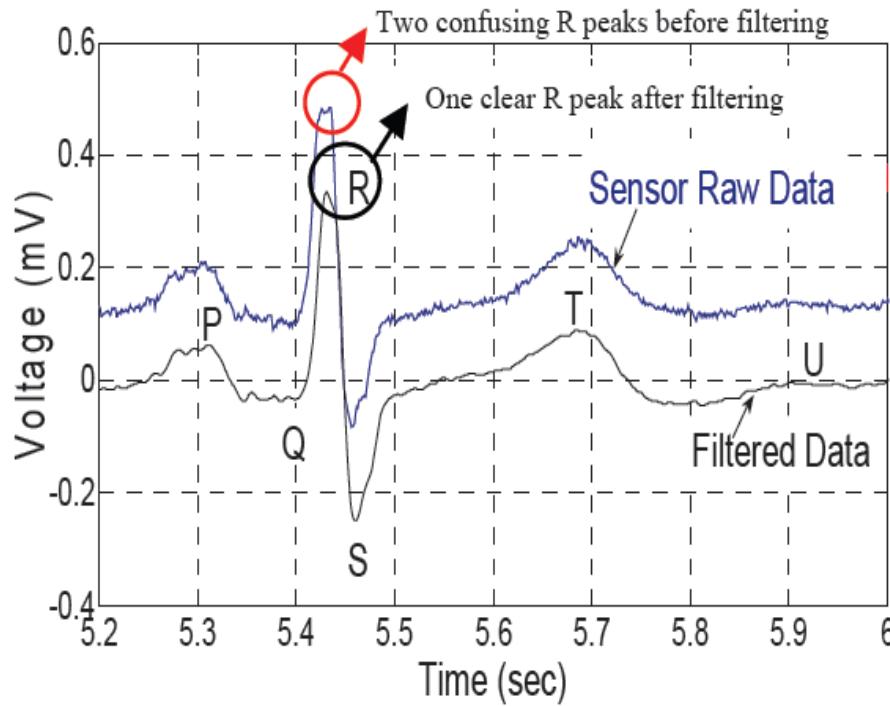
## A Shannon-inspired SEC-based Subthreshold ECG Processor in 45nm CMOS

Rami Abdallah and Naresh R. Shanbhag  
[Abdallah, Shanbhag, JSSC, Nov. 2013]  
Dept. ECE, University of Illinois at Urbana-Champaign



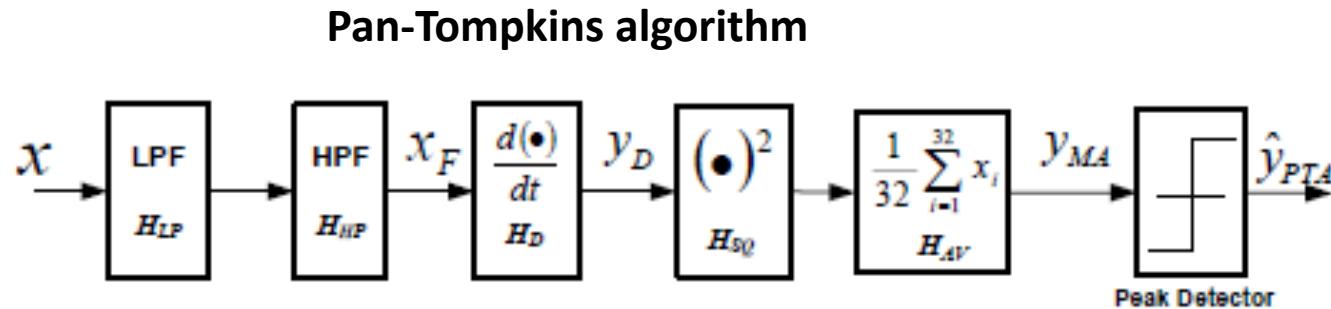
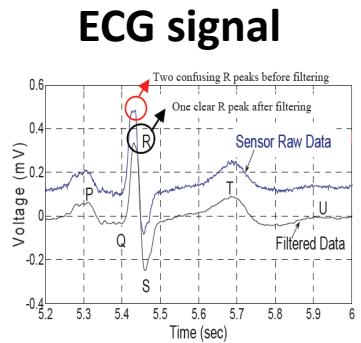
Implantable/wearable  
Biomedical devices

# ECG Analysis



- Cardiovascular disease detection
  - Real-time QRS detection
  - Beat-to-beat (RR) extraction

# System Requirements & Information-based Metrics



**Binary decision**  
1: abnormal  
0: normal

$10b - 14b, 0.1\text{kHz} - 1\text{kHz}$

## Application-level Information-based Metrics

**sensitivity**

$$Se = \frac{TP}{TP + FN} = p_{tp}$$

↑

(set of +ve inputs)

**positivity**

$${}^+P = \frac{TP}{TP + FP} = 1 - p_{fp}$$

↑

(set of +ve outputs)

American Heart Association

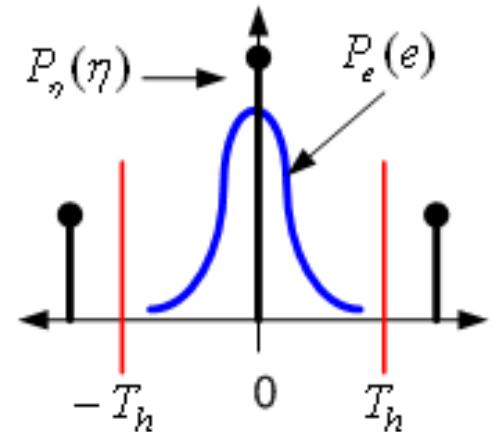
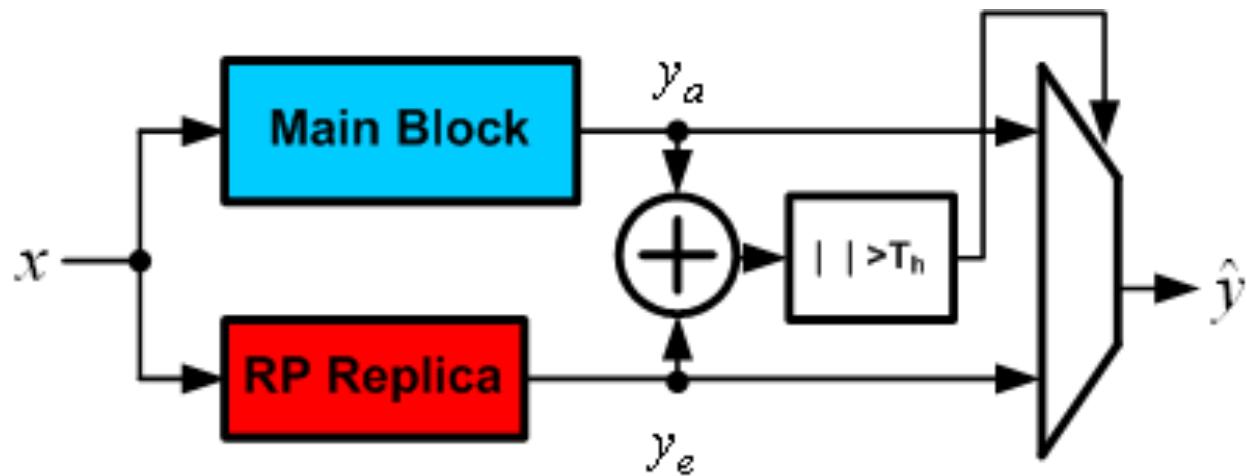
**requirements**

$$Se, {}^+P > 95\%$$

**H/W error rate**

$$p_\eta = \Pr\{\text{observing an error in a clock cycle}\}$$

# Reduced Precision Replica

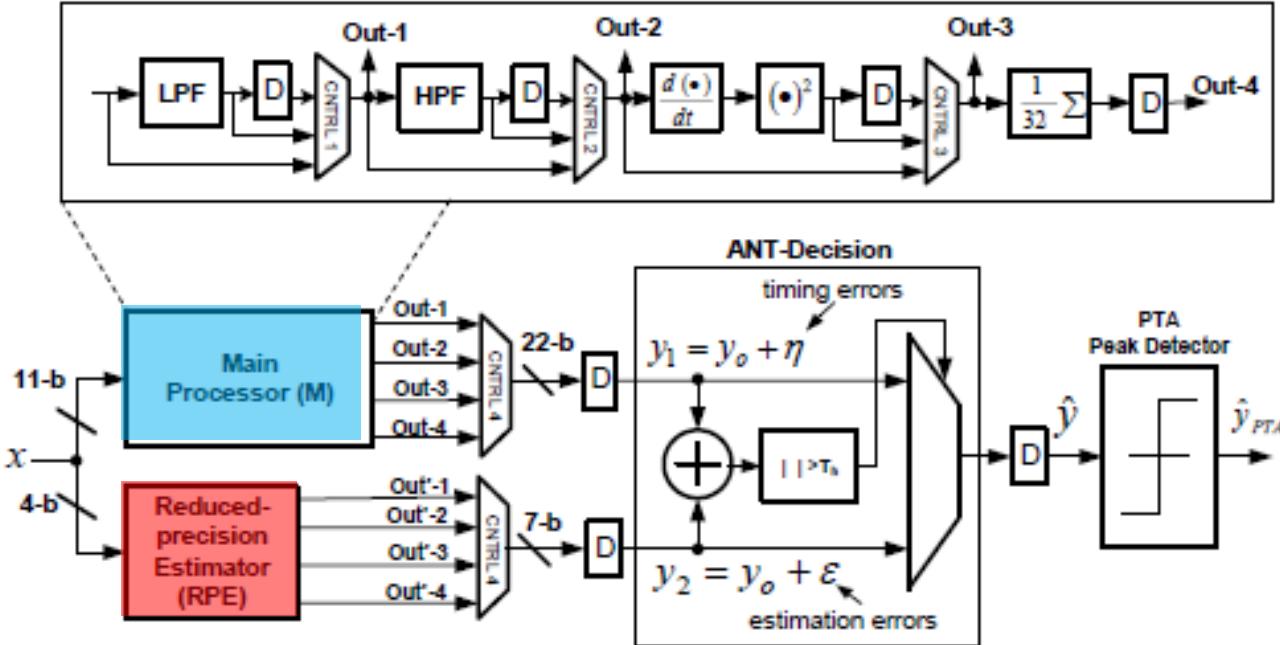


- employ a reduced-precision version of main block
- estimation error  $y_e$  equals quantization noise (bounded)

$$T_h = \max_{\forall x} [y_o[n] - y_e[n]]$$

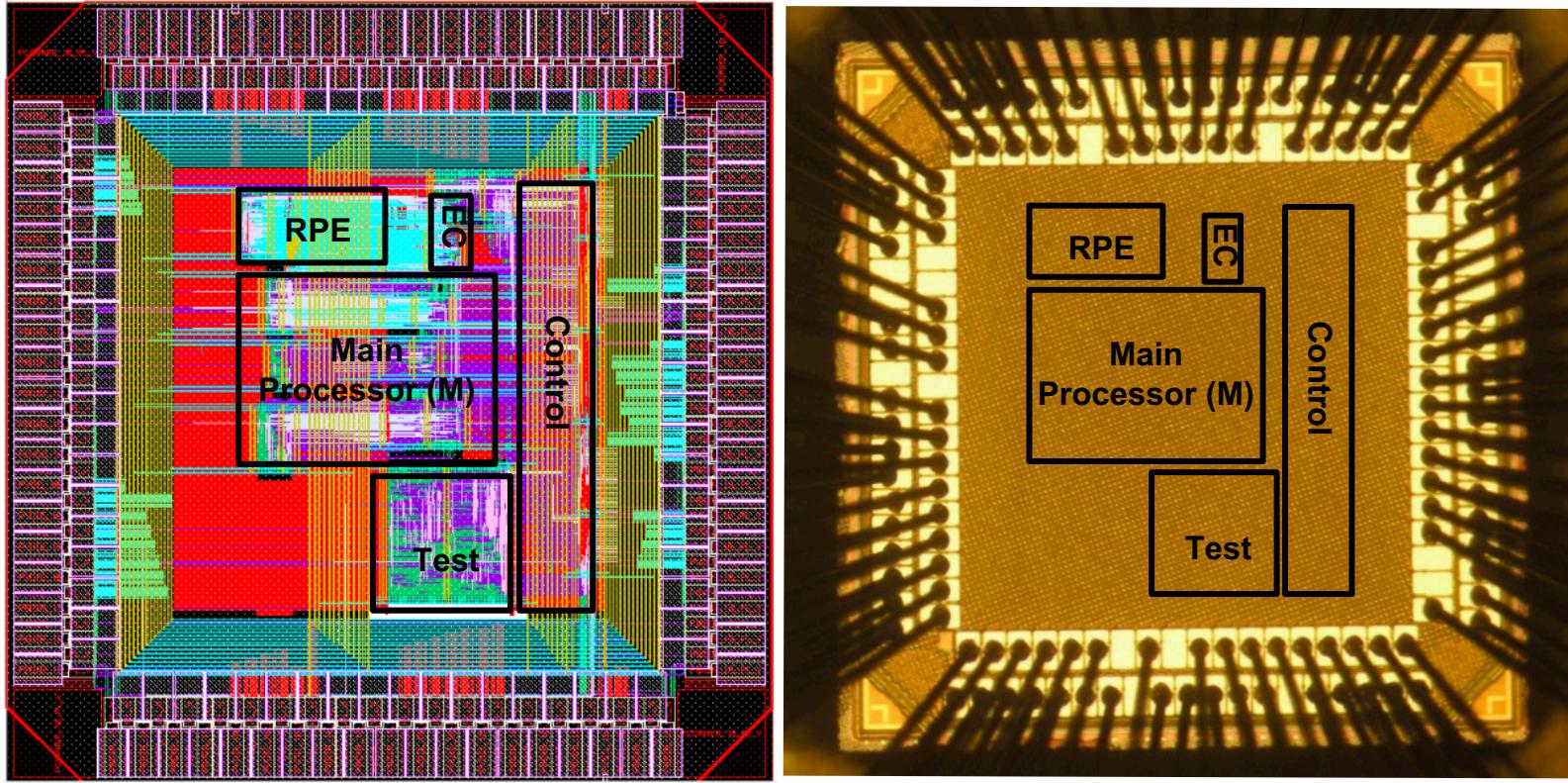
$$P(|\eta[n] - e[n]| < T_h \mid \eta \neq 0) = 0 \quad \text{false alarm probability} = 0$$

# ECG Processor Architecture



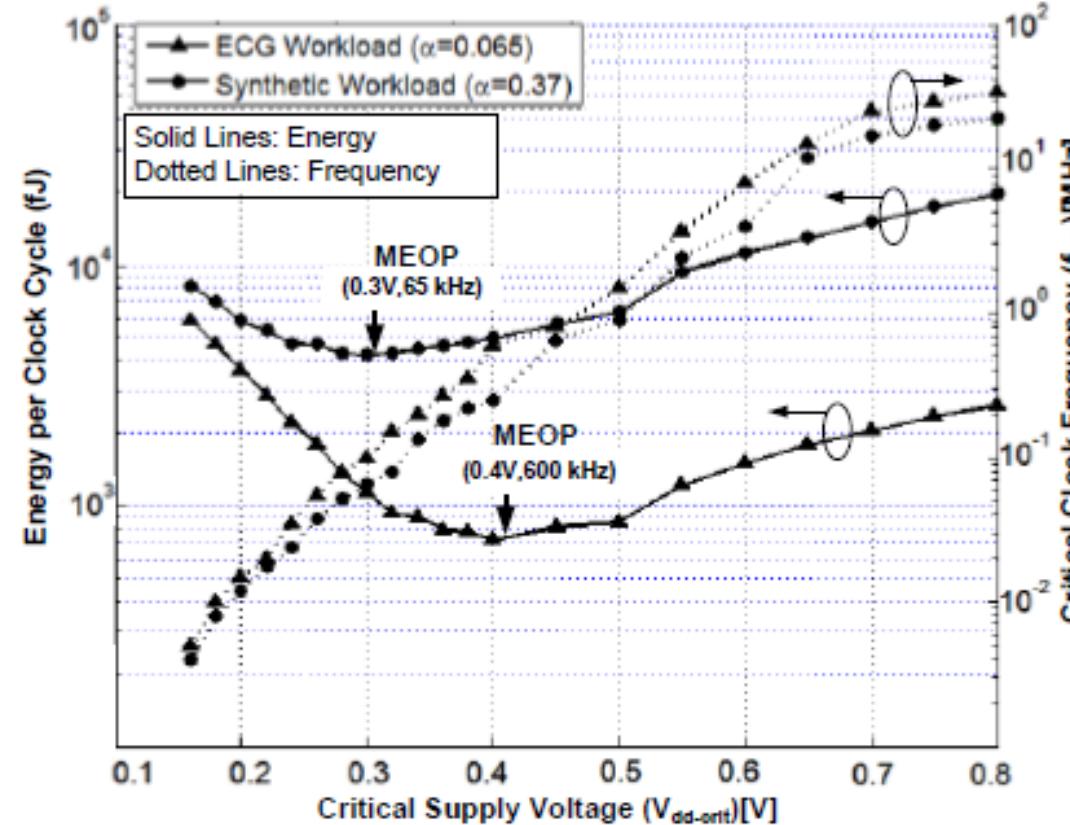
reconfigurable – controlled  
error locations via pipelining  
latch placements

Algorithmic noise-tolerance (ANT) – Reduced precision redundancy (RPR)  
(logic overhead < 30%)



- IBM 45nm SOI using ARM standard cell library
- Minimum-strength cells for low power and error-friendly slack
- 12 power domains
- Gate complexity: 36K NAND2 Equivalent Gate
- Chip area: 1.25mm x 1.3mm

# Error-Free Operation



- tested with 2 workloads with different switching activity
- MIT-BIH arrhythmia dataset conventional MEOP: 0.7pJ, 0.4V, 600kHz
- synthetic workload conventional MEOP: 4pJ, 0.3V, 65kHz

# Error-Rate ( $p_\eta$ )

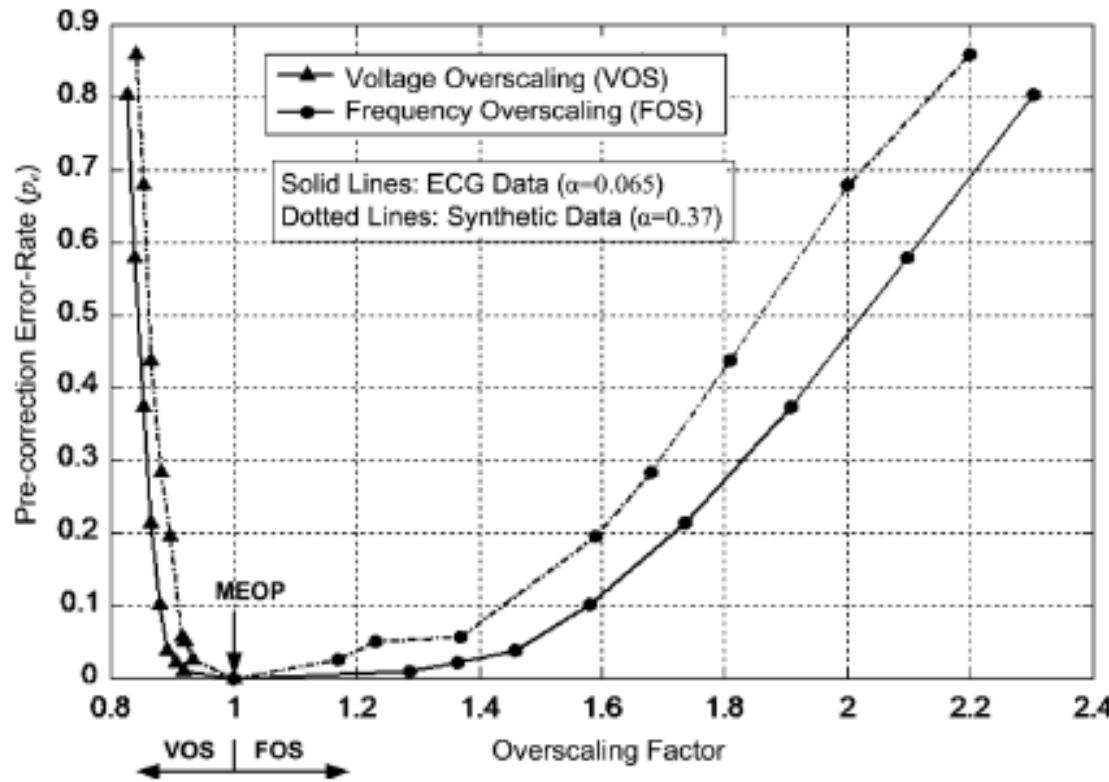


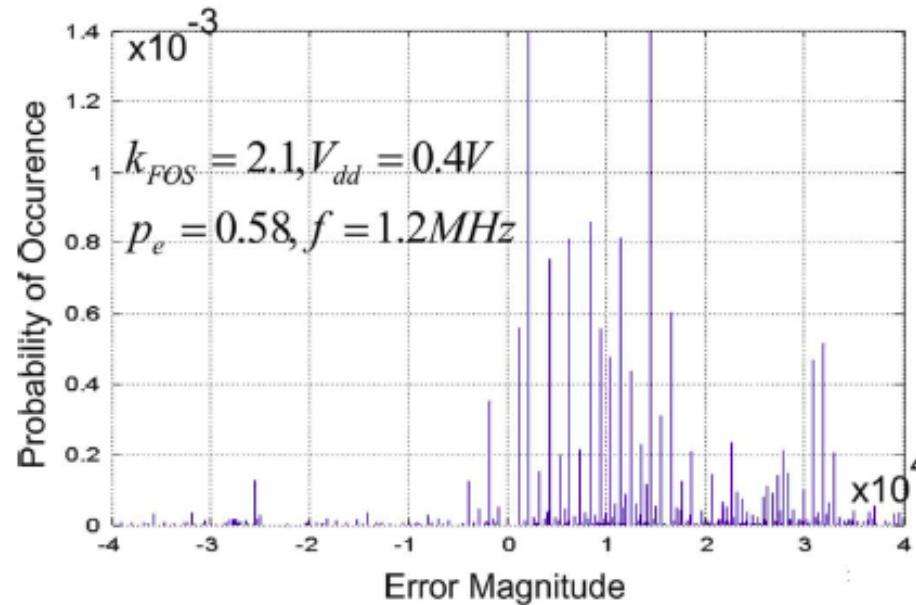
Fig. 9. Measured pre-correction error rate of ECG processor under voltage overscaling and frequency overscaling.

**minimum energy operating point  
(MEOP)**

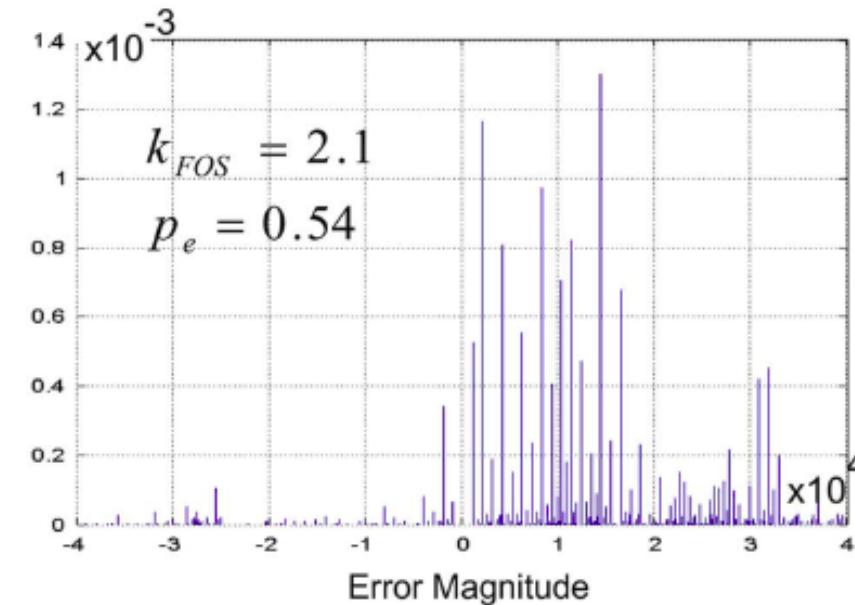
- M-block output
- error rate more sensitive to VOS than to FOS (frequency overscaling)

# Error PMF ( $P(\eta)$ )

measured

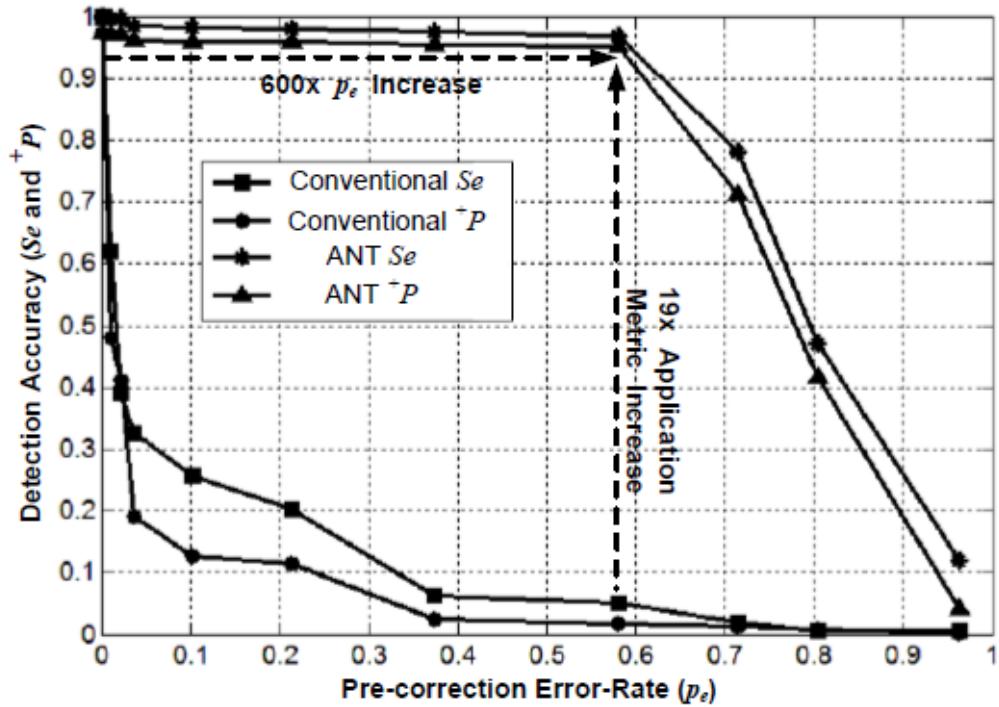


simulated

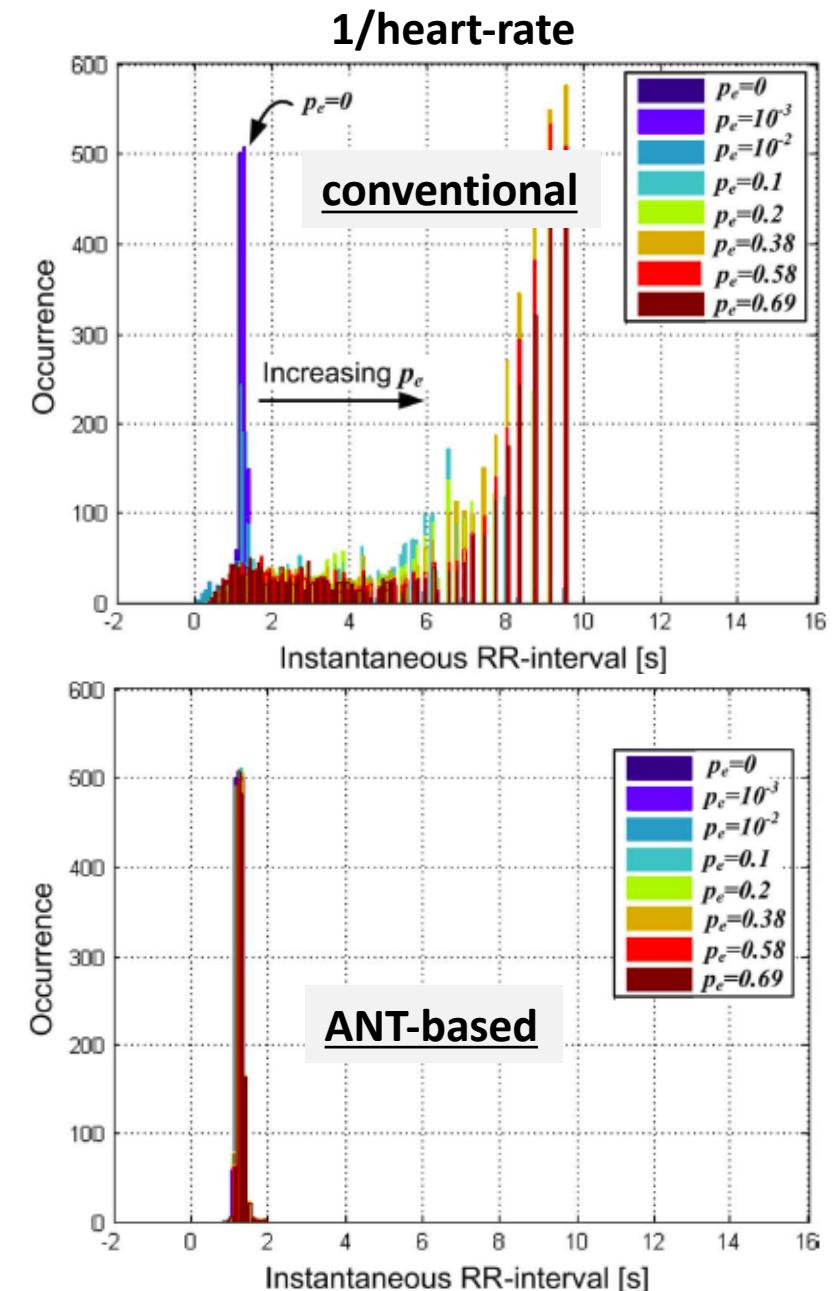


- M-block (MA block) output
- measured and back-annotated gate-level simulations match → implications for other SEC techniques

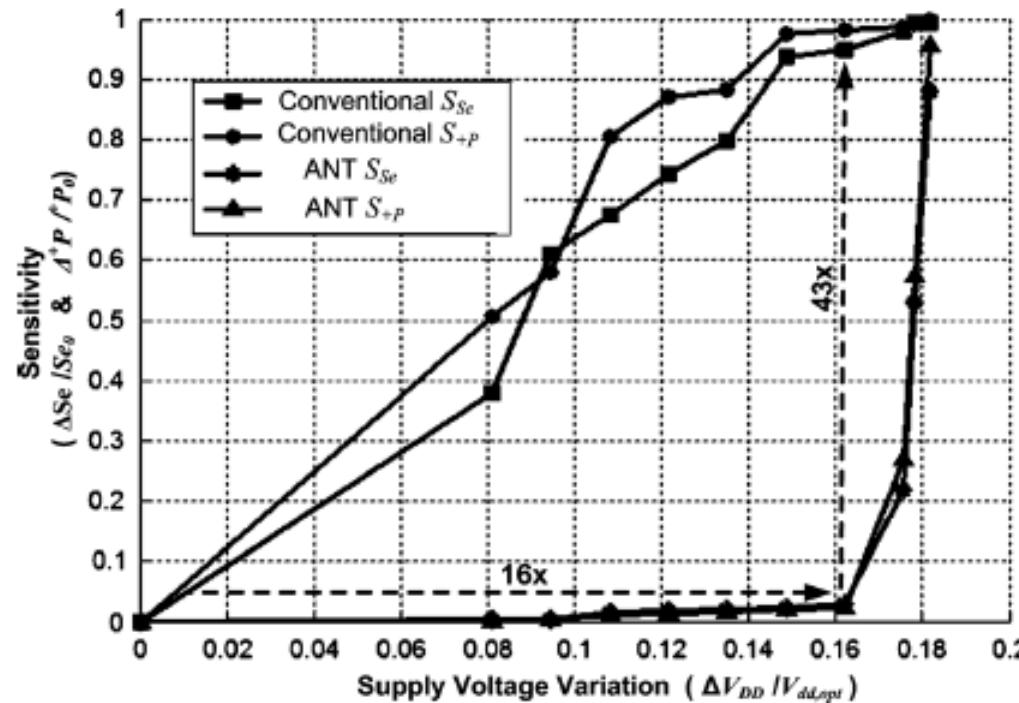
# Robustness to Errors



600X greater  $p_\eta$  (raw error-rate) handling capability  
 $error-rate = 58\%$

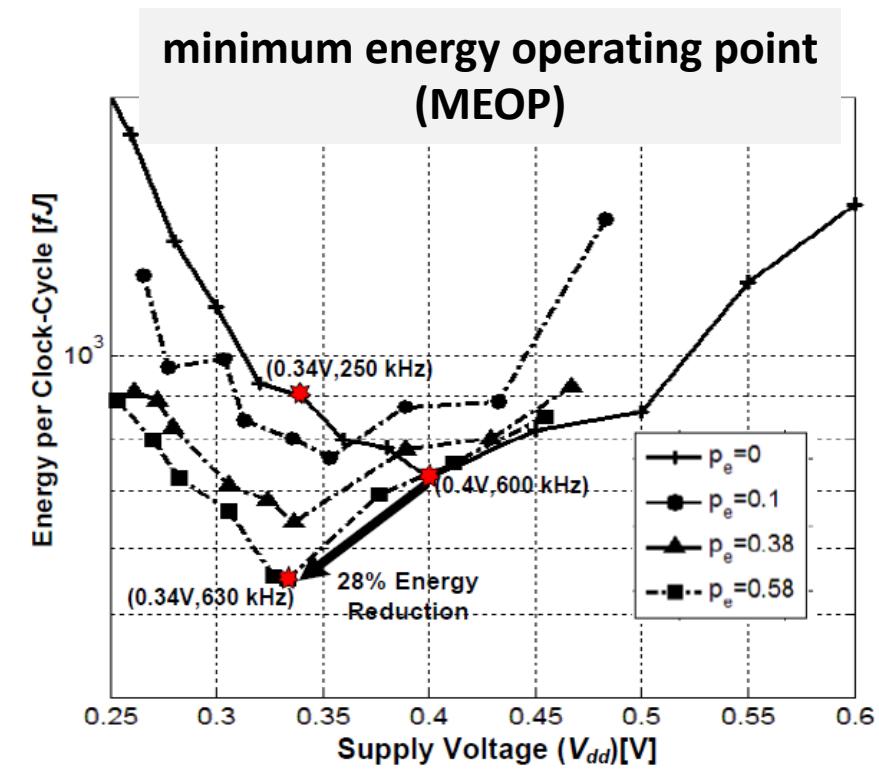


# Robustness to Voltage Variations



tolerates 16X higher voltage variations about MEOP (0.4V)  
43X less metric sensitivity

# Energy Savings @ MEOP



*raw error-rate = 58%*  
28% reduction of energy @ MEOP & 15% higher throughput

# Comparison & Conclusions

	Ashouei (IMEC '11)	Sridhara (TI '10)	Blaauw (U-Mich)	Bowman (Intel)	This Work
<b>Technology</b>	90 nm	130nm	45nm	65nm	45nm
<b>Operating Point</b>	1 MHz 0.4 V	7 kHz 0.5 V	185 MHz 1.16 V	3 GHz 1.0 V	0.6 MHz 0.34 V
<b>Energy/Cycle</b>	13pJ	29pJ	505pJ	N.A.	0.52pJ
<b>Energy/ (Cycle x 1Kgates)</b>	<b>68fJ</b>	<b>483fJ</b>	8416 fJ	N.A.	<b>15fJ</b>
<b>Raw Hardware Error Rate (pe)</b>	0	0	<b>0.04</b>	<b>0.001</b>	<b>0.58</b>
<b>Energy Savings (past PoFF)</b>	0	0	5%	7%	28%

- A 0.34V with 56% hardware error rate ECG processing chip using statistical error compensation
- 28% energy saving benefit past point of first failure (PoFF)
- 4.7X more energy-efficient than state-of-the-art

# Case Study - III

## SSNOC

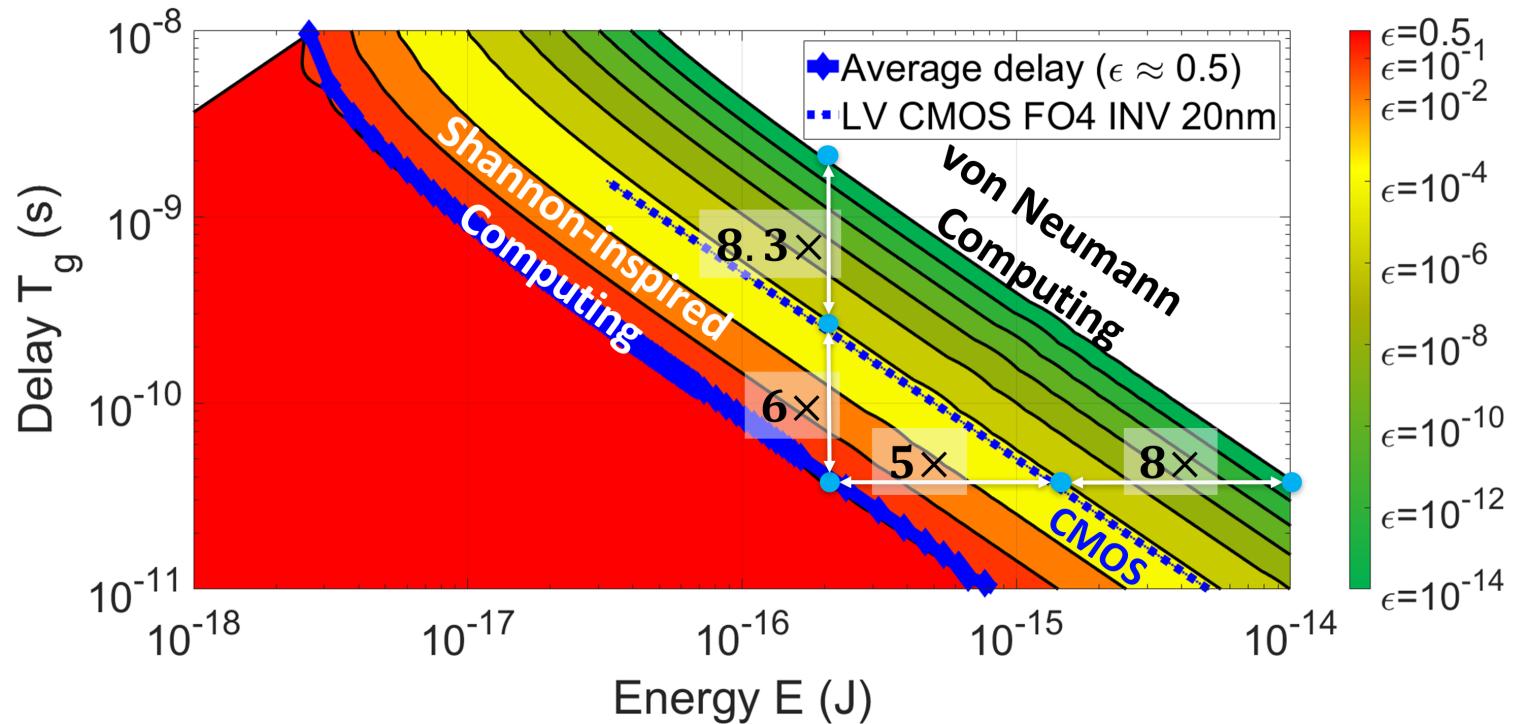
Eric Kim and Naresh R. Shanbhag

Dept. ECE, University of Illinois at Urbana-Champaign

[XXX, JSSC, Nov. 2013]

# Shannon-inspired Spintronics

# All Spin Logic (ASL) Energy-Delay Challenge



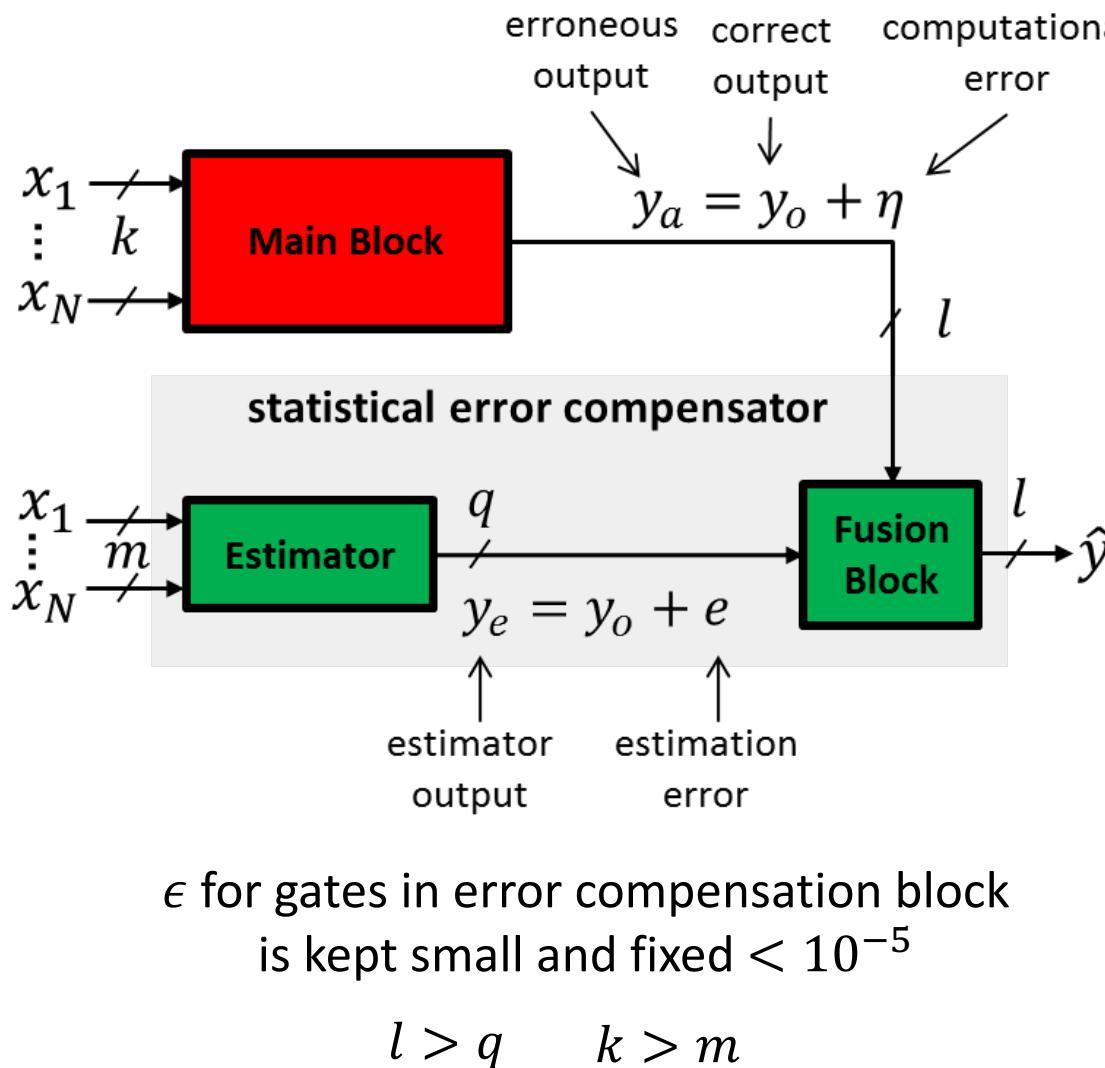
- STT devices need 3-4 OOM > charge for switching vs. CMOS [Ganguly, JxCDC'17]
- need to operate ASL at high error rates (~1%) to beat CMOS
- making it incompatible with von Neumann architectures
- **Can Shannon-inspired model of computation help?**

- applying Shannon-inspired Computing to **all-spin logic** is very challenging → all gates make errors with high probability
- idea → use energy-delay-error rate trade-off to **shape error statistics**

$$\epsilon \propto \exp\{-\alpha \sqrt{E T_d}\}$$

Joint work with Intel: Sasi Manipatruni, Dmitri Nikonov, and Ian Young

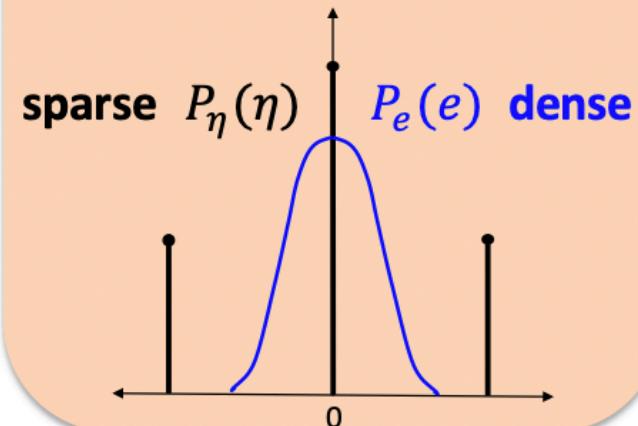
# Shannon-inspired Computing for ASL



## Fusion block

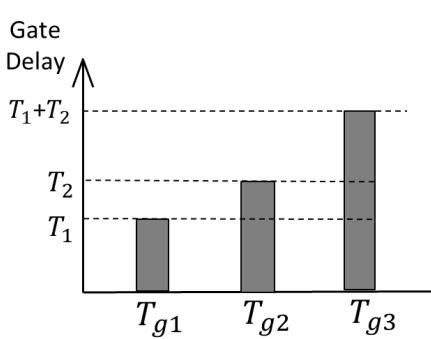
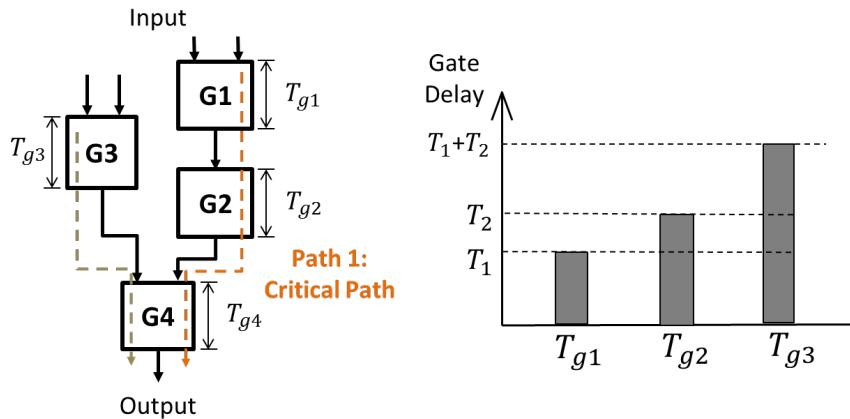
$$\begin{aligned}\hat{\eta} &= \text{ML estimate of } \eta \\ \hat{y} &= y_a - \hat{\eta}\end{aligned}$$

## Required error PMF property: disparity



# Engineering Sparse Error Distributions

## Intra-Path Delay Balancing (IPDB)

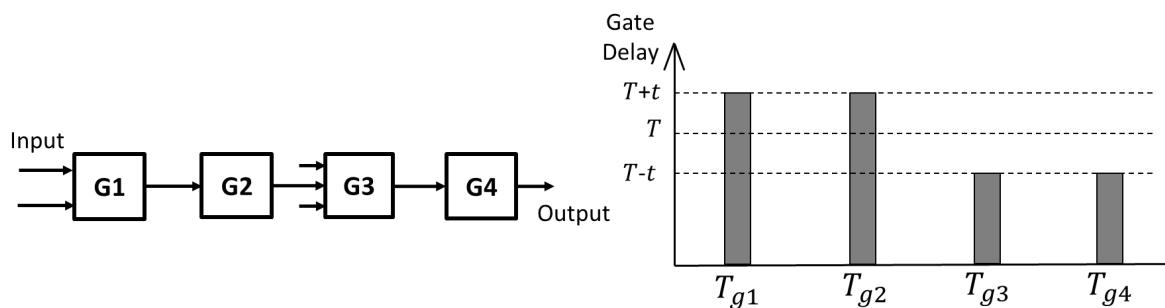


“maximally” slow network → “minimally” error-prone network, **without energy increase**

### IDPB property:

each gate is on **at least one critical path**, and all **critical paths have identical delays**

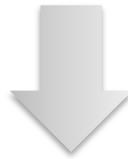
## Inter-Path Delay Redistribution (IPDR)



preserves IPDB property  
**generates sparse error PMF**

# Error Distribution Engineering

All gate  
delays equal

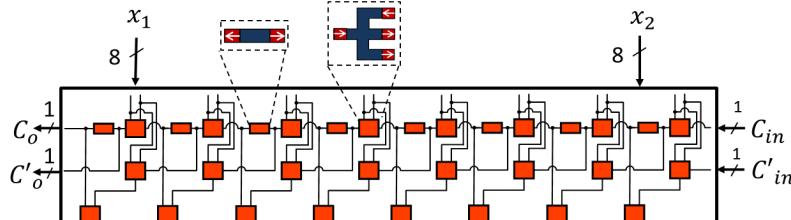


After IPDB

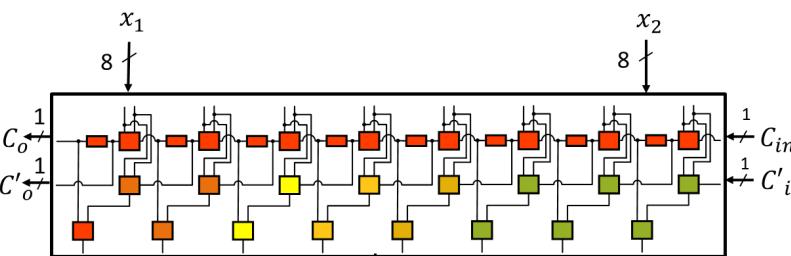


After IPDB  
and IPDR

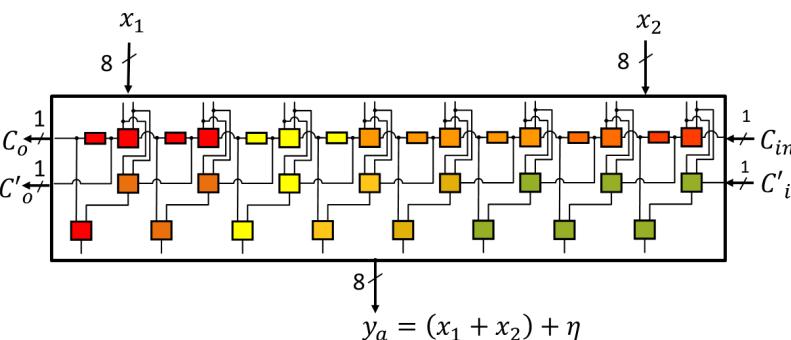
8b RCA



$$y_a = (x_1 + x_2) + \eta$$

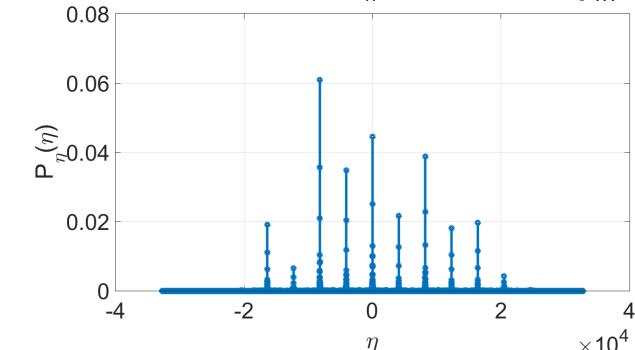
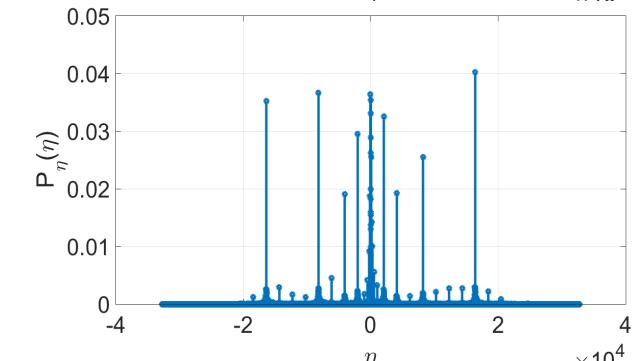
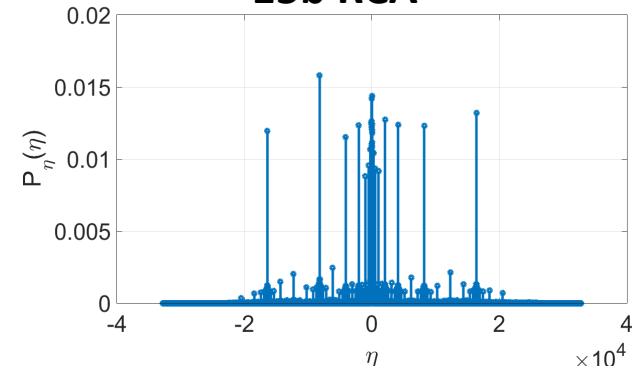


$$y_a = (x_1 + x_2) + \eta$$



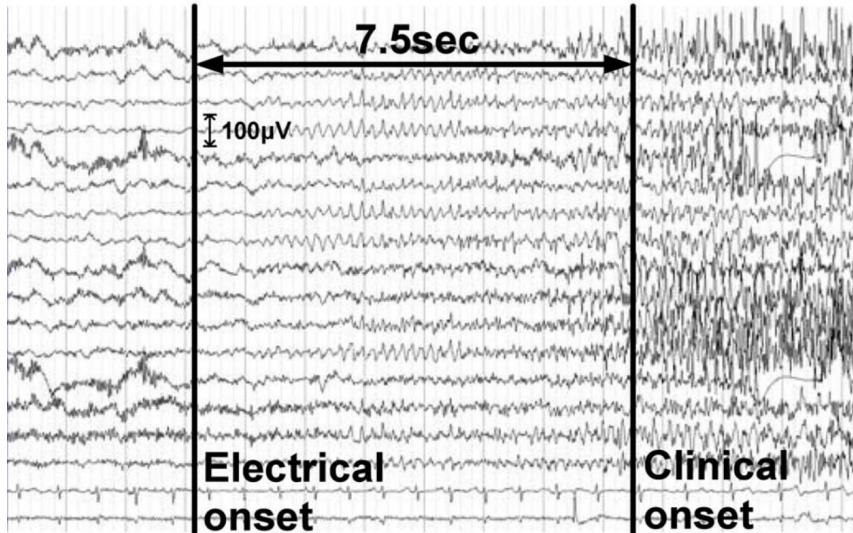
$$y_a = (x_1 + x_2) + \eta$$

15b RCA

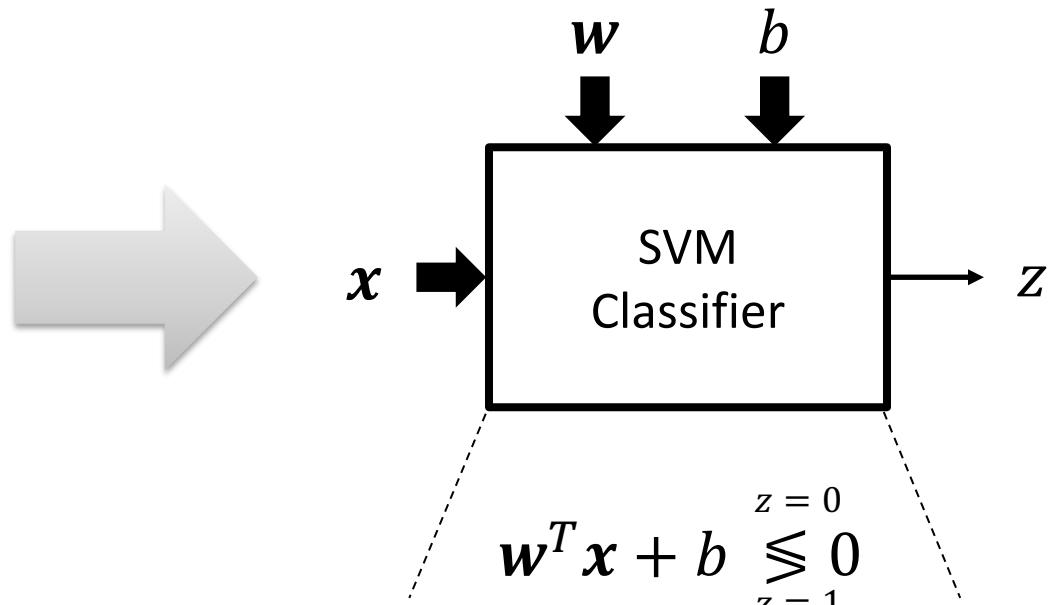


# Seizure Detection using a Support Vector Machine (SVM)

CHB-MIT EEG dataset



[Verma-JSSC-2010]

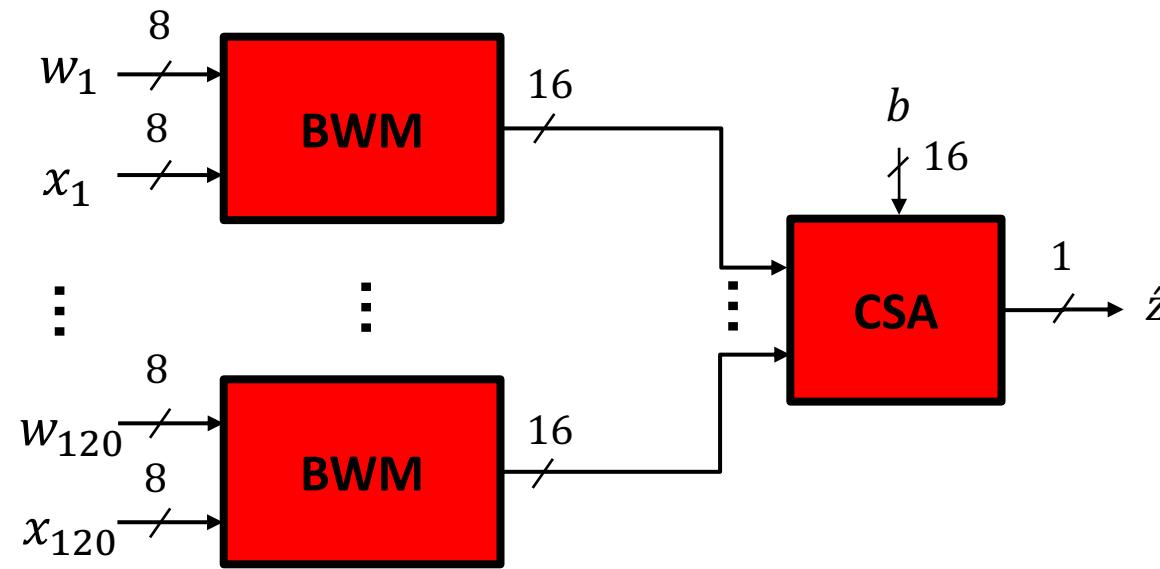


$x$ : feature vector extracted from EEG signals

$w$ : trained weight vector       $b$ : trained scalar bias

$z$ : decision;  $z = 1 \Rightarrow$  seizure

# Conventional SVM Architecture

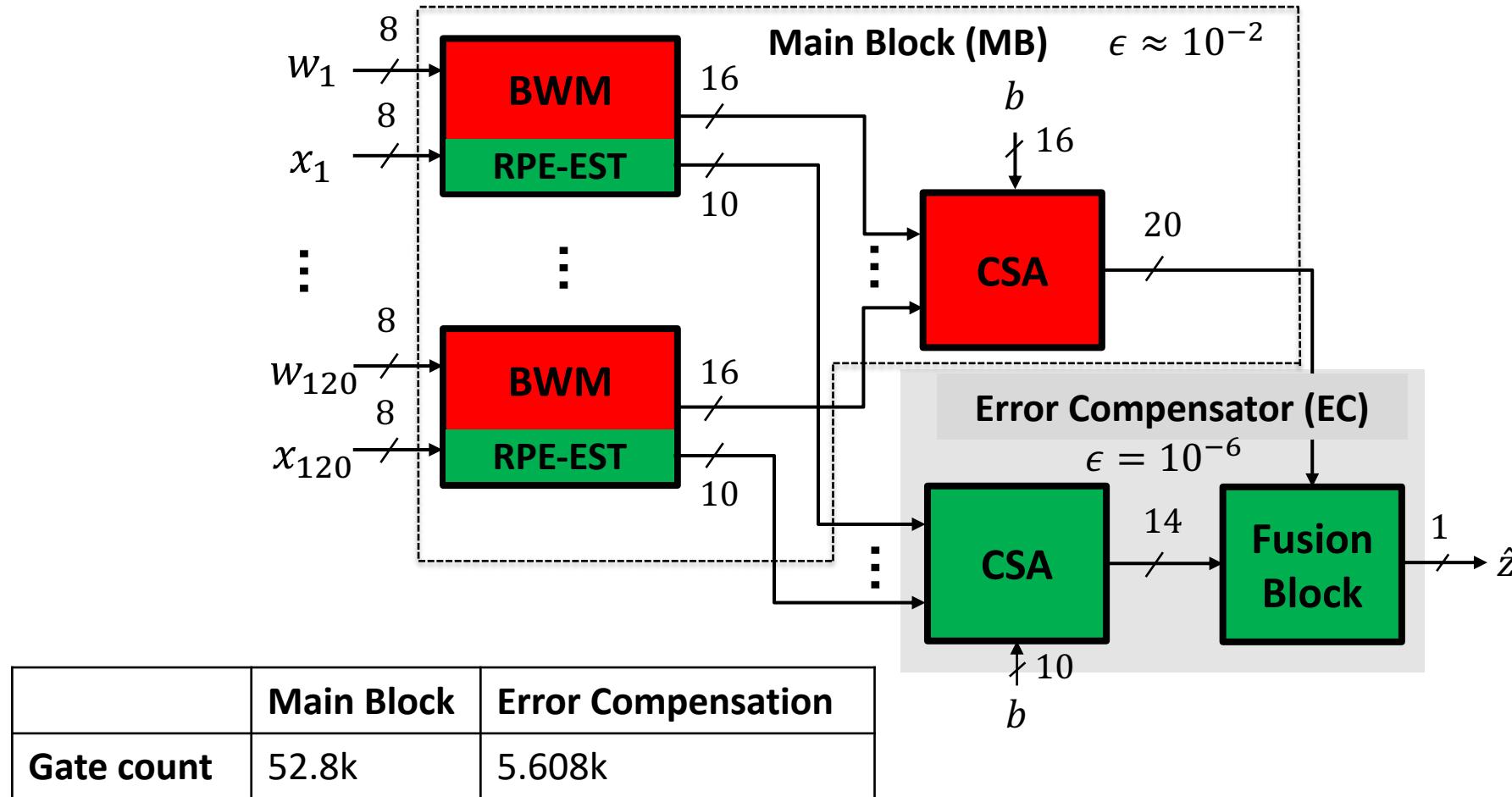


BWM: Baugh-Wooley Multiplier

CSA: Carry Save Adder

Total gate count: 52.8k

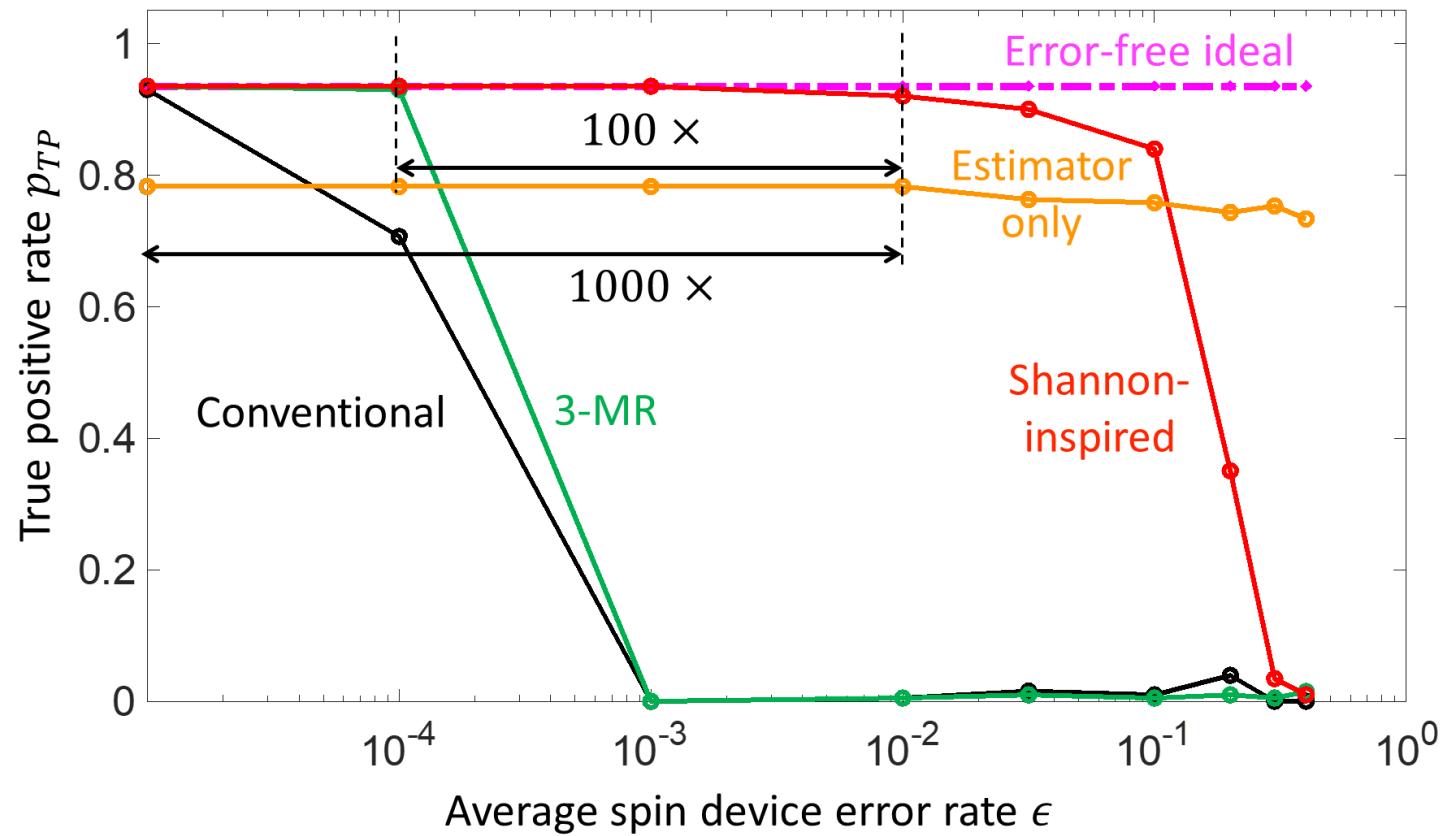
# Shannon-inspired SVM Architecture



**RPE-EST:** Reduced precision embedded estimator

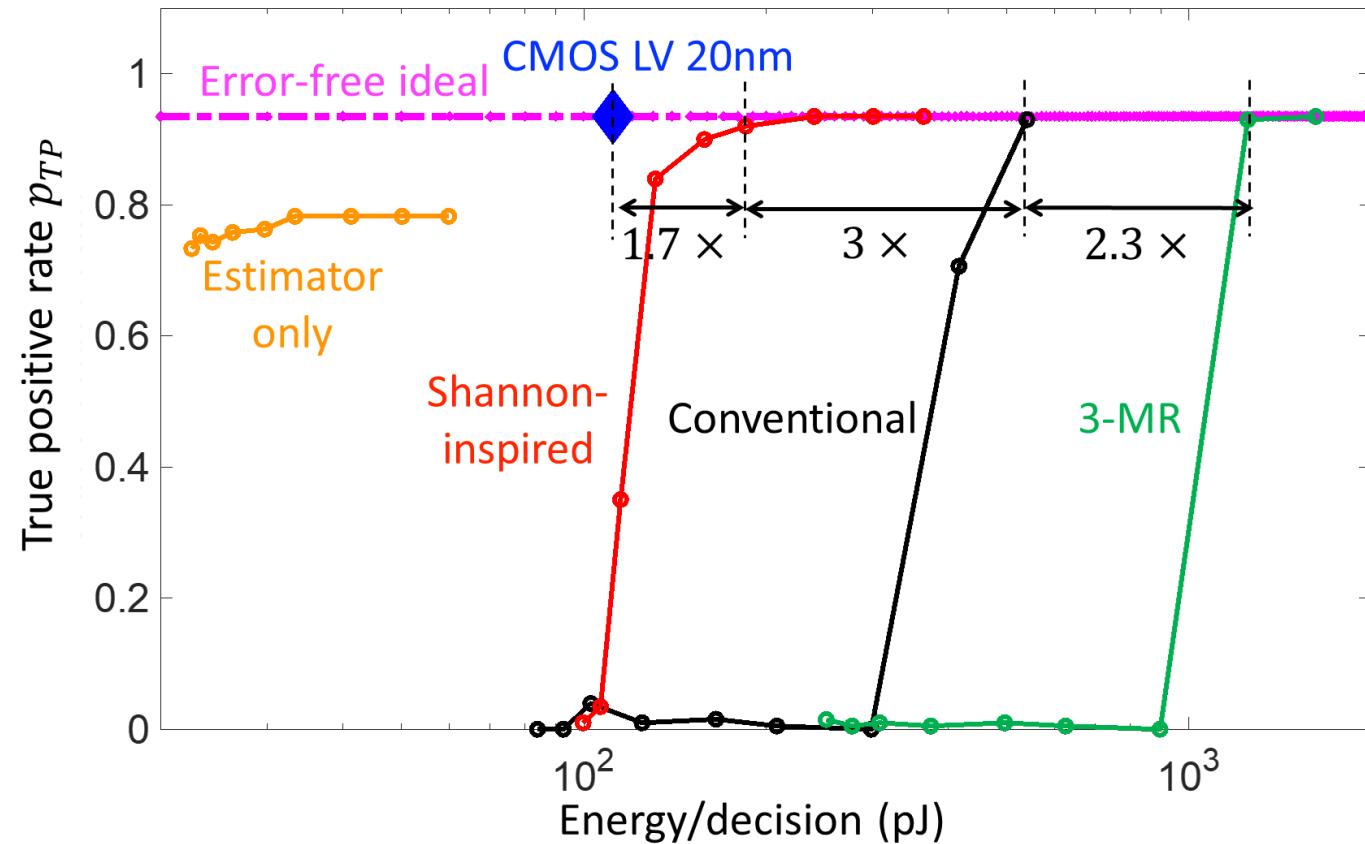
(11% overhead)

# Accuracy vs. Device Error Rate



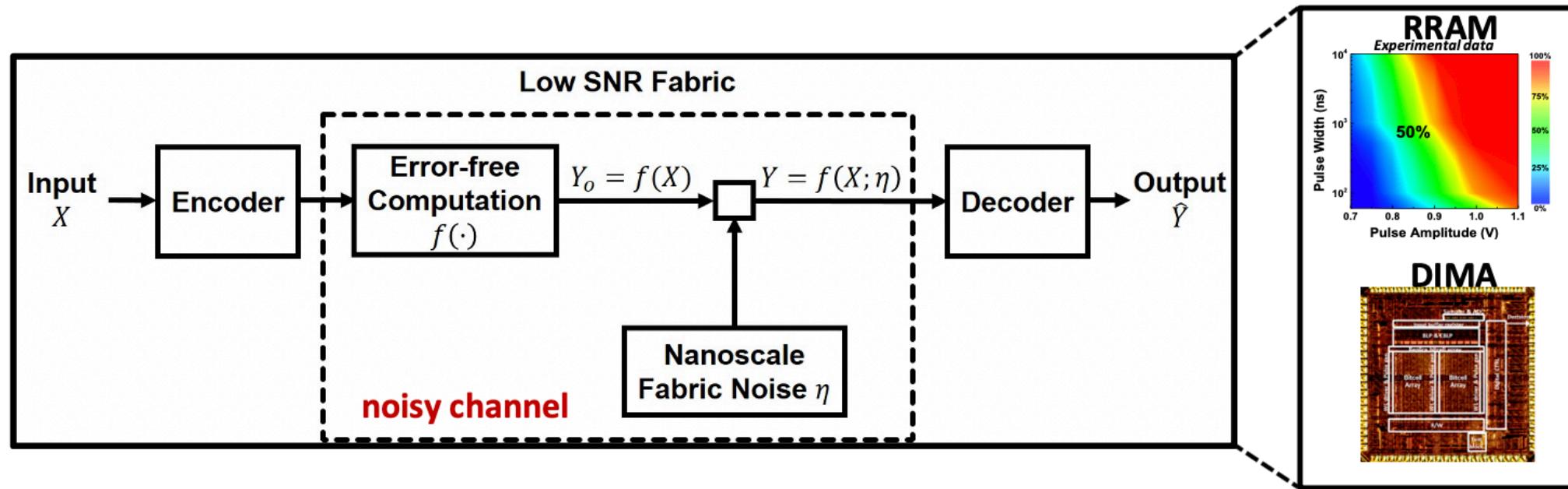
- false positive rate = 1%
- tolerates 1000× improvement in average device error rate

# Accuracy vs. Decision Energy



- Shannon-inspired methods offer 100X greater error compensation
- with 7X lower energy than fault-tolerance methods *yet cannot beat CMOS*

# Shannon-inspired Model of Computing



- use information-based metrics e.g., mutual information  $I(Y_o; \hat{Y})$
- design low SNR fabrics - RRAM, MRAM, **deep in-memory architecture (DIMA)**, quantum
- develop statistical error-compensation (SEC) techniques



J. Von Neumann, *Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components*, Princeton University Press (1956)

“treatment of error is ***unsatisfactory and ad hoc***”

“***error should be treated as information has been***, by the works of  
**C. E. Shannon**”

“The present treatment ***falls short of achieving this***”

## Course Web Page

<https://courses.grainger.illinois.edu/ece598nsg/fa2020/>

<https://courses.grainger.illinois.edu/ece498nsu/fa2020/>

<http://shanbhag.ece.uiuc.edu>