GDR BIOCOMP Summer School
Roscoff, June 2017

Computing with Unreliable Nanodevices

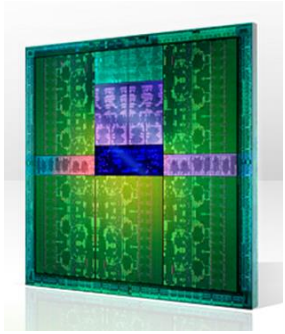
Damien Querlioz

Chargé de recherche CNRS, https://sites.google.com/site/damienquerlioz/

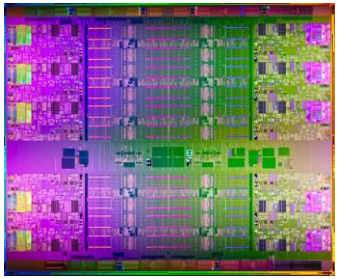
Centre de Nanosciences et de Nanotechnologies
(ex-Institut d'Electronique Fondamentale)
Univ. Paris-Sud, Université Paris-Saclay, CNRS, ***Orsay***

Microelectronics has achieved a lot

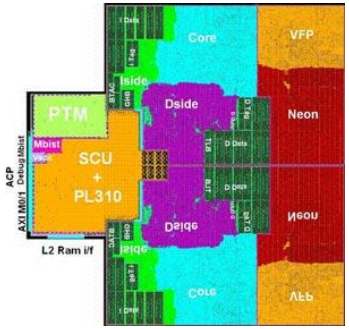
- **Right now** we can now incredible things…



High Perf graphic processor (GPU)
Nvidia TESLA K10
>7 billions transistors
>Tflops on a single chip !

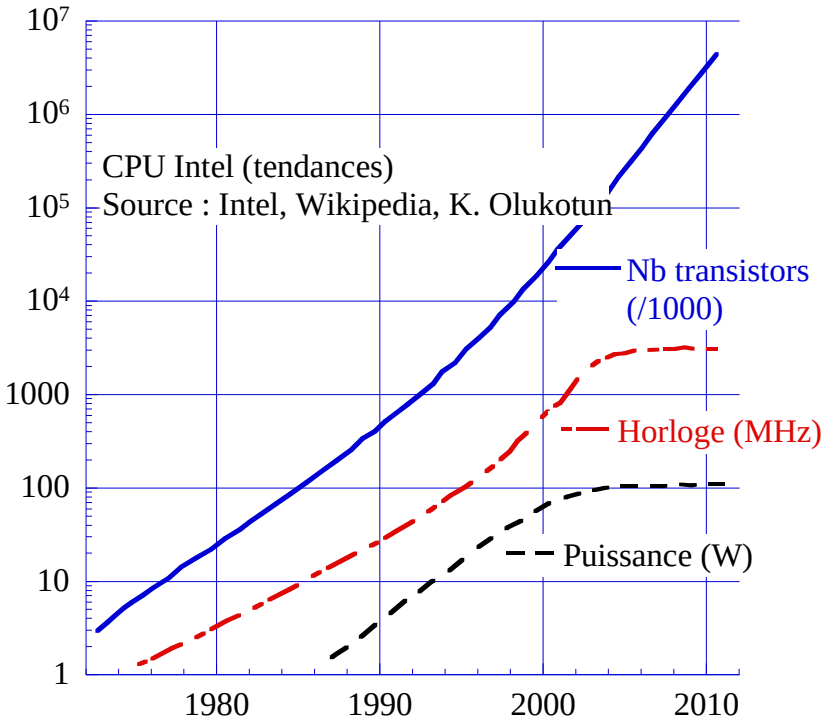


High Perf microprocessor (CPU)
Intel Xeon E7
>2 billions transistors
100W



Mobile CPU and GPU
Based on ARM designs
3D gaming with <1W !

How this happened



On a single chip,
all transistors work

But things are changing

Microelectronics « crisis »

April 2015

- Since transistors have entered the nanometer regime, transistor scaling brings little benefit

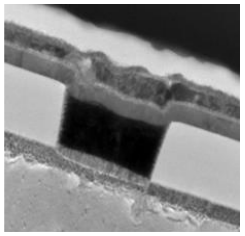
- Very difficult to make nanotransistors reliable

The end of the traditional model of microelectronics

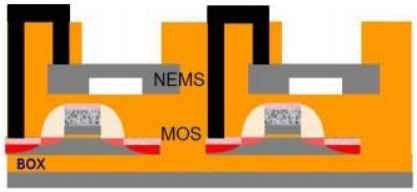
« More than Moore » research

- Nanotechnology brings new devices to CMOS that can give it new features

➢ Novel functions: memory effects, sensor, energy harvesting…



Novel memories
CEA LETI

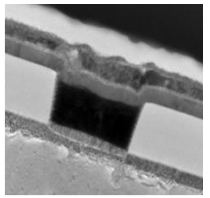


NEMS sensors

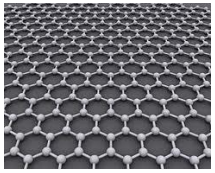
More than Moore comes with challenges

Electronic nanodevices : the good and the challenge

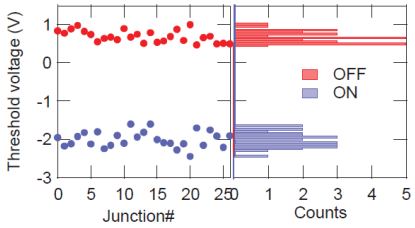
- Nanoelectronics' devices are amazing
  - Compact, low power
  - Novel functions: memory effects, sensor, energy harvesting…



CEA LETI



- But…
  - **Variability**
  - Noise, Faults



Borghetti et al, Nature 464 (2010)

- Few products with nanoelectronics!

Can we design for technology's imperfections?

Biology: the nanoelectronics champion?

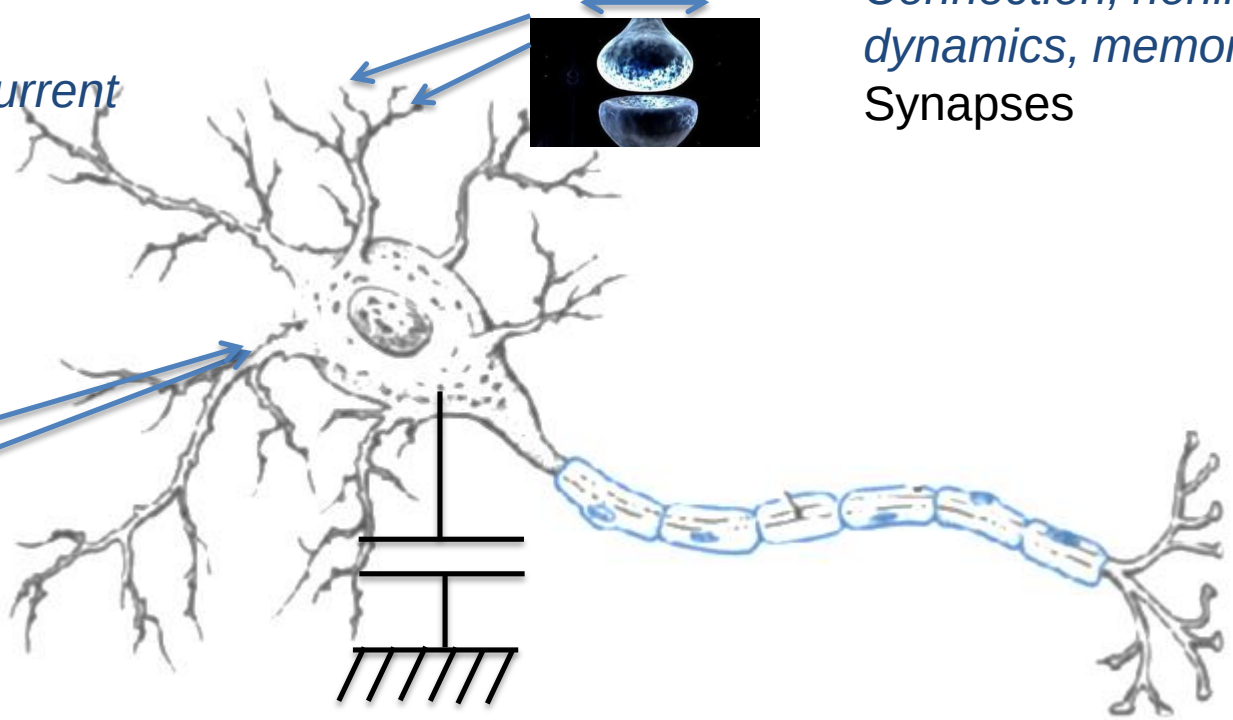
Neuron compute using nanoelectronics/nanoionics devices

~500nm

Connection, nonlinear dynamics, memory
Synapses

Controlled current source
Ion channel

~5nm

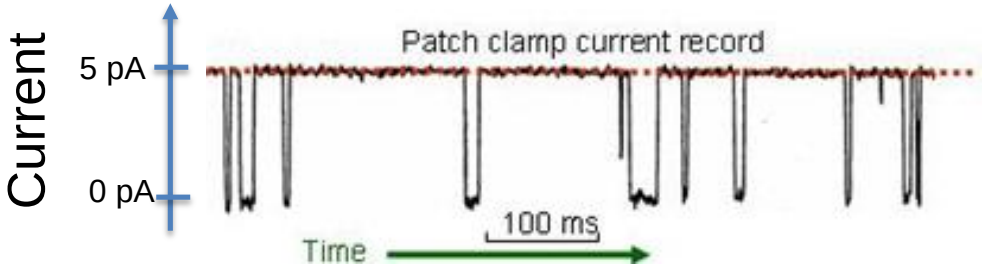


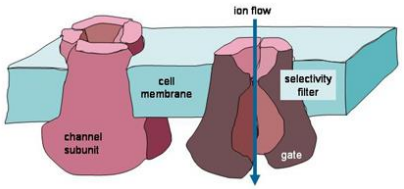
Brains are extremely energy efficient based on nanodevices. Can it be an inspiration?

Biology's nanoelectronics

- Biological « nanodevices » are **not** ideal electron devices

Example of ion channel (controlled current source)





Hille et al, 2001

Current is pure telegraph noise!

And in a neuron, only ~100 ion channels open at the same time (Schneidman, 1998)

Total ion channel current fluctuates by $\sqrt{100}/100 = 10\%$!

Can we take some inspiration from the way biology manages to compute with "imperfect" devices?

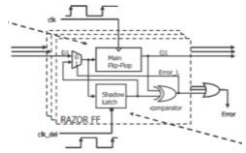
Question of the class

- We struggle with unreliable nanodevices, while biology shows us it is possible to strive with them

This class:
Can we also compute reliably with awesome but imperfect nanodevices?

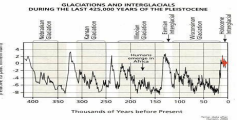
Outline of the class

- The reliability issue of nanodevices

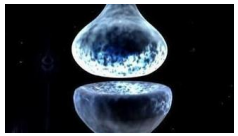


- Computing with errors: “Detect and correct”

- Approximate computation



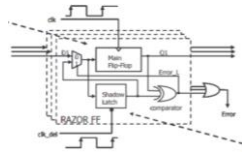
- Computing with noise



- Toward *bioinspired* computing

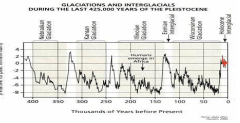
Outline of the class

- The reliability issue of nanodevices

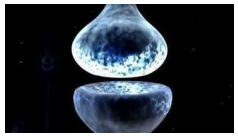


- Computing with errors: “Detect and correct”

- Approximate computation



- Computing with noise



- Toward *bioinspired* computing

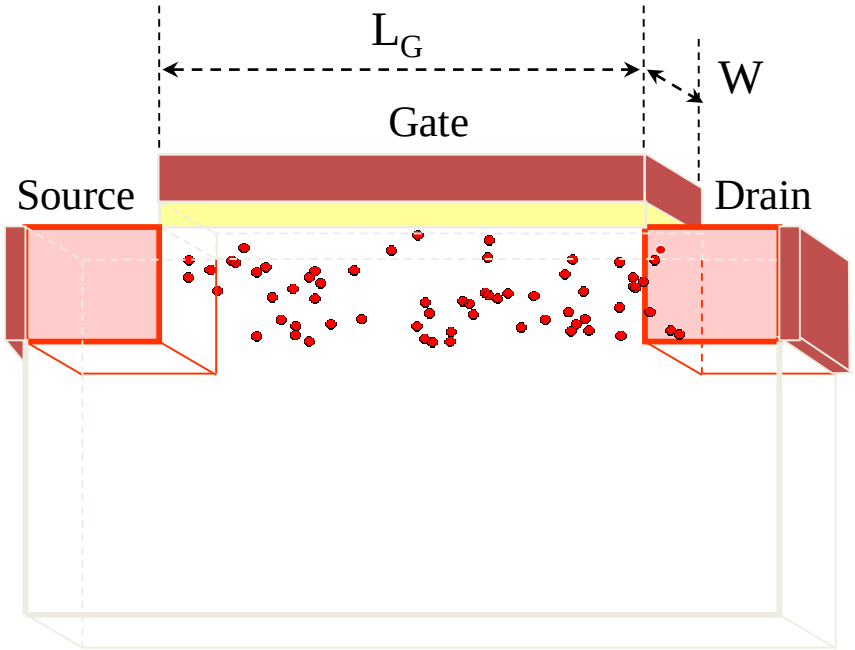
Nanodevices are imperfect

- Variability
- Noise / unpredictability

First, let's look at the case of CMOS

NanoCMOS: high variability becomes intrinsic

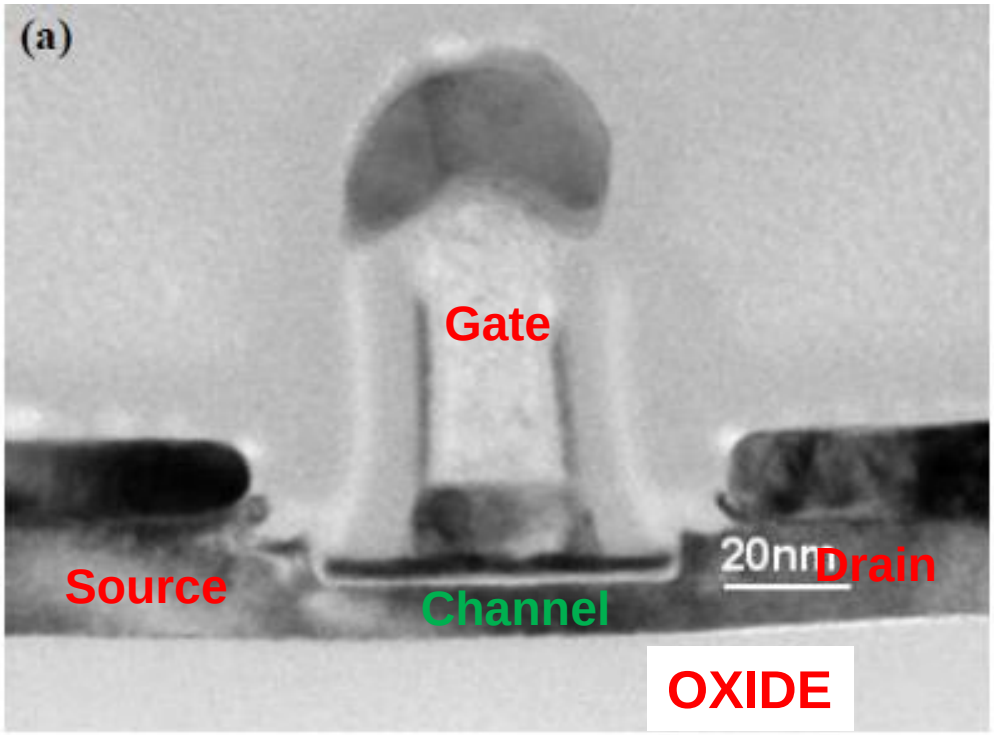
- Only a few dozens of dopants in the channel



Variation in number of dopants causes unavoidable mismatch between transistors

Fully Depleted SOI (FDSOI)

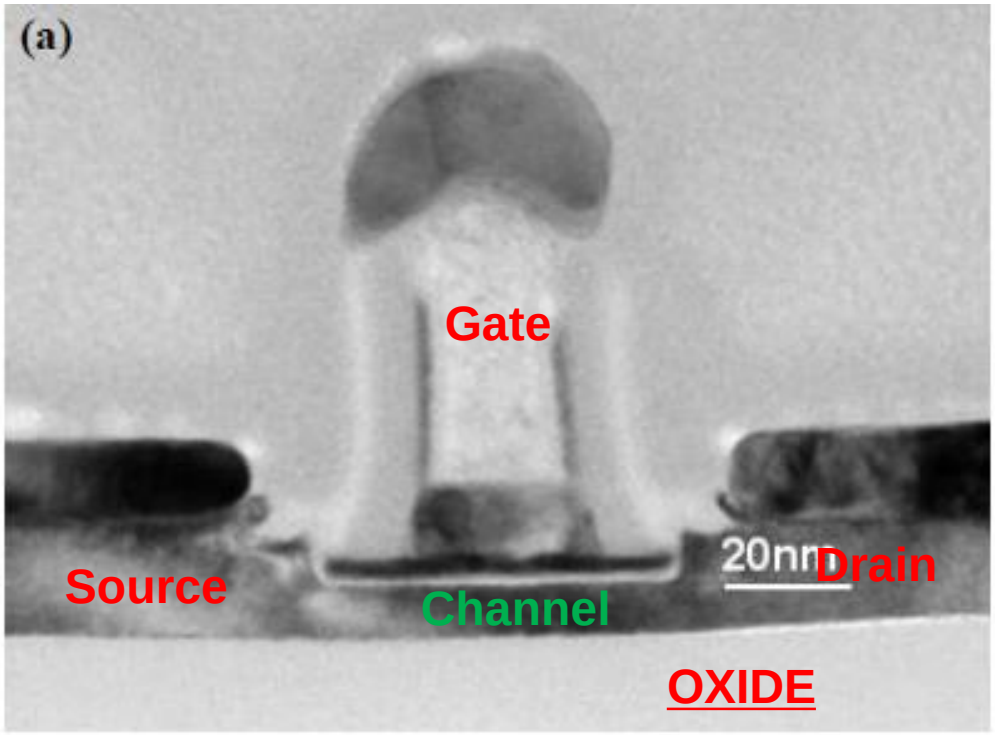
- ST Microelectronics since 28nm
- Now Samsung and Globalfoundries



Silicon on Insulator wafer

Weber et al, IEDM 2008

Fully Depleted SOI (FDSOI)



Channel is
created physically

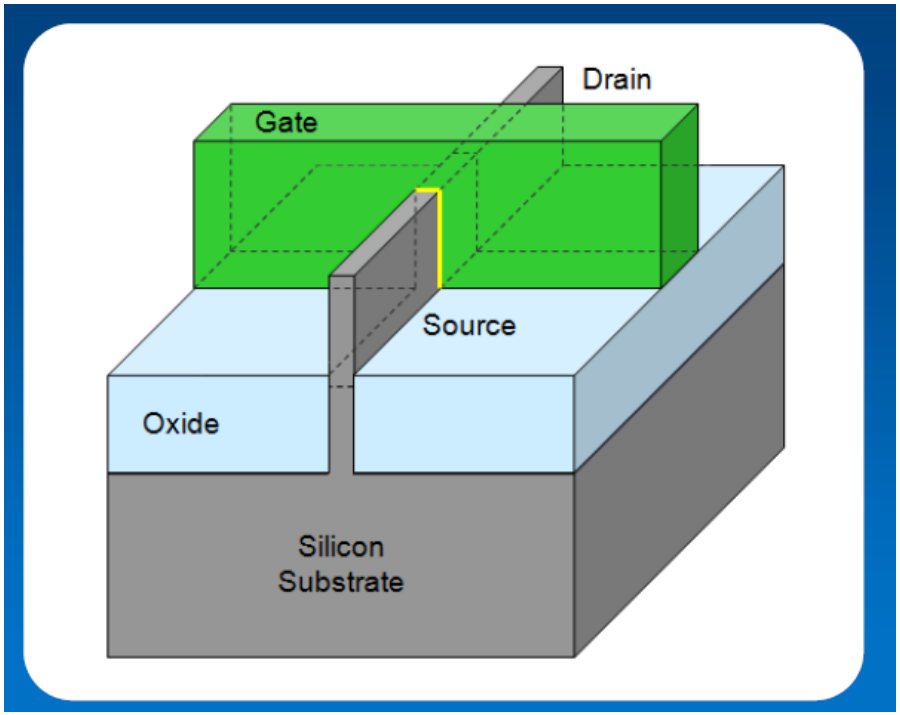
- Low leakage

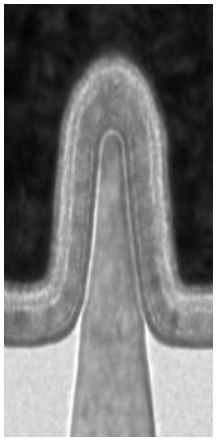
- No need for
channel doping!

Weber et al, IEDM 2008

The other option : FinFET

- Intel: since 22nm
- Also TSMC, Samsung, Gloablfoundries…





Burried channel surrounded by gate!

Undoped channel

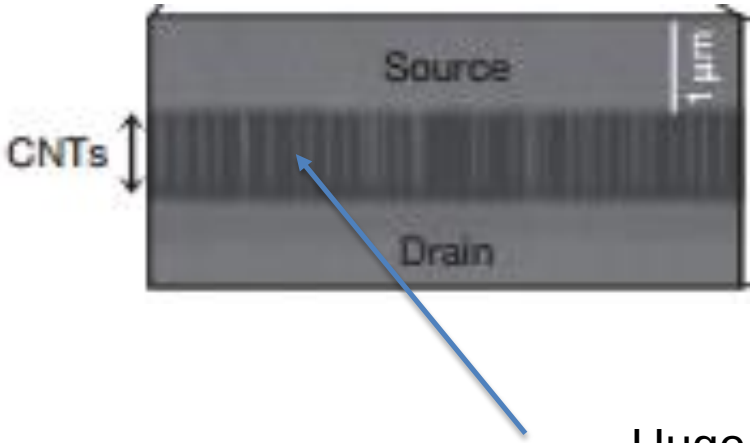
These solutions still have limitations

- New variability issues when ultrascaled
- (gate material…)

How about true nanoelectronics ?

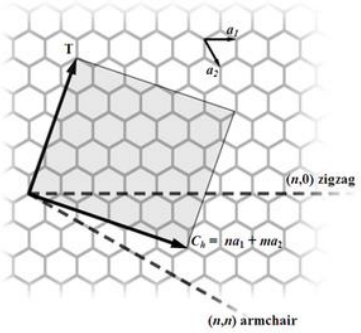
Carbon nanotubes FETs 1/2

- Variability issue of carbon nanotubes



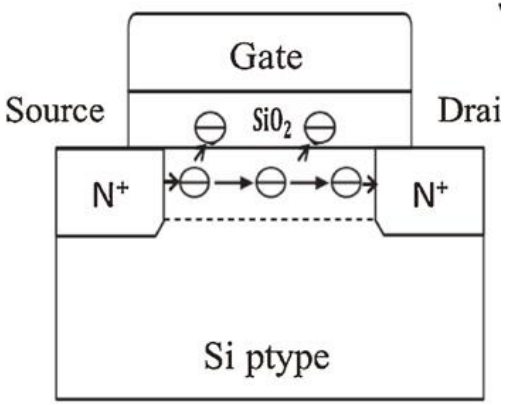
Shulaker et al ,
Nature 2013

Huge **intrinsic** variety of carbon nanotubes
Sorting techniques inaccurate!

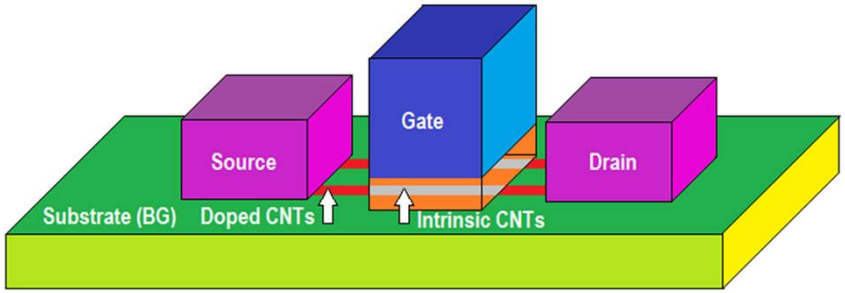


Carbon nanotubes FETs 2/2

- Low frequency noise order of magnitudes higher than bulk materials!
- Noise issue:
  - Surface vs. volume

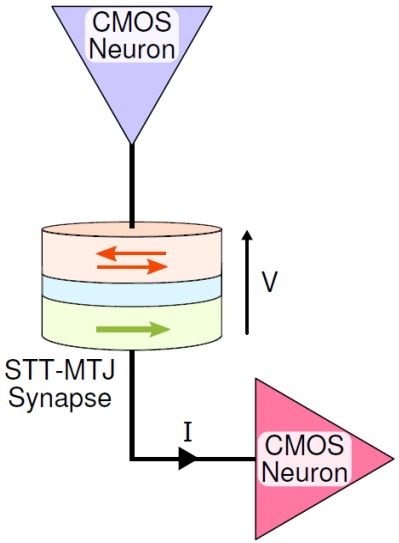


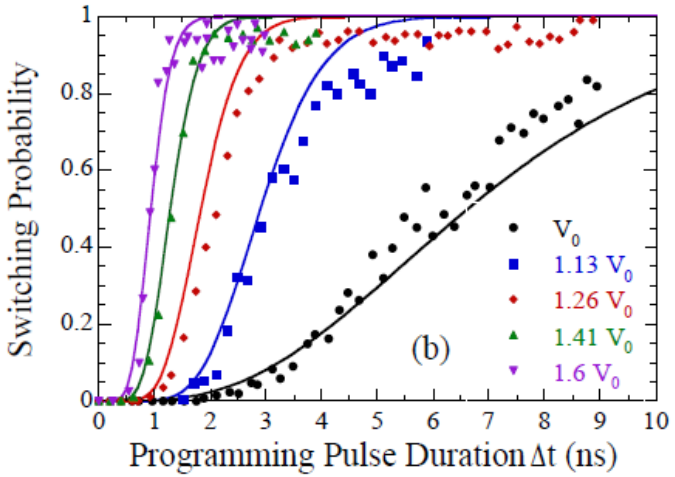
Bulk case



Nano case

Stochastic switching in Spin Torque Magnetic Memory





Measurements on an in-plane magnetization structure (T. Devolder, IEF/Univ. Paris-Sud)

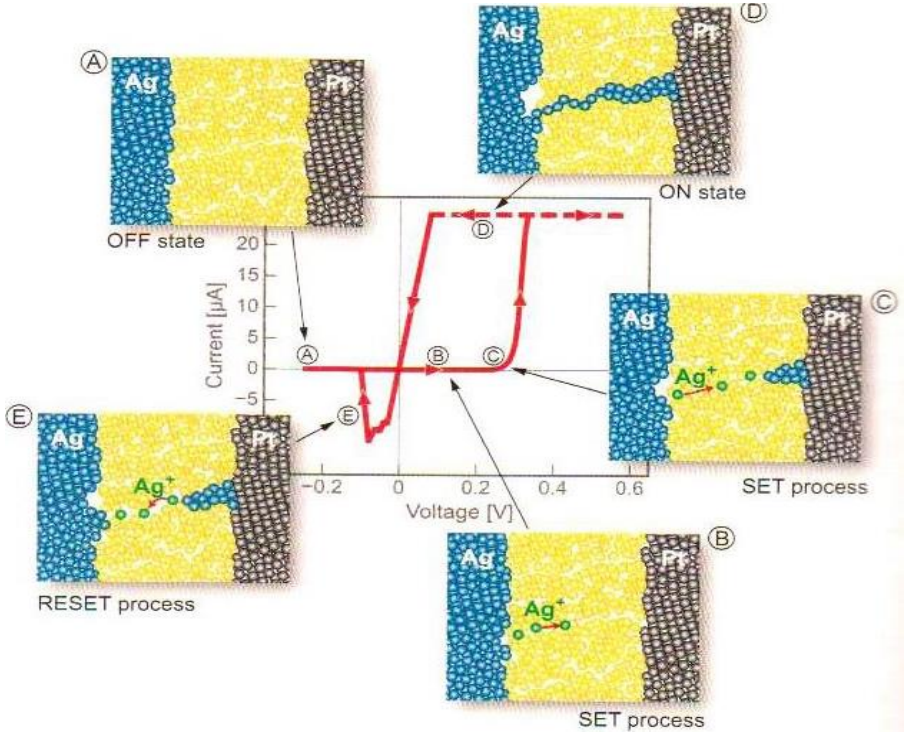
V_0 : arbitrary unit (<1V)

- Switching fundamentally stochastic due to basic magnetism physics (no switching without thermal noise)

Vincent et al, IEEE Trans Biomed Circuits and Systems, 9, p. 166, 2015

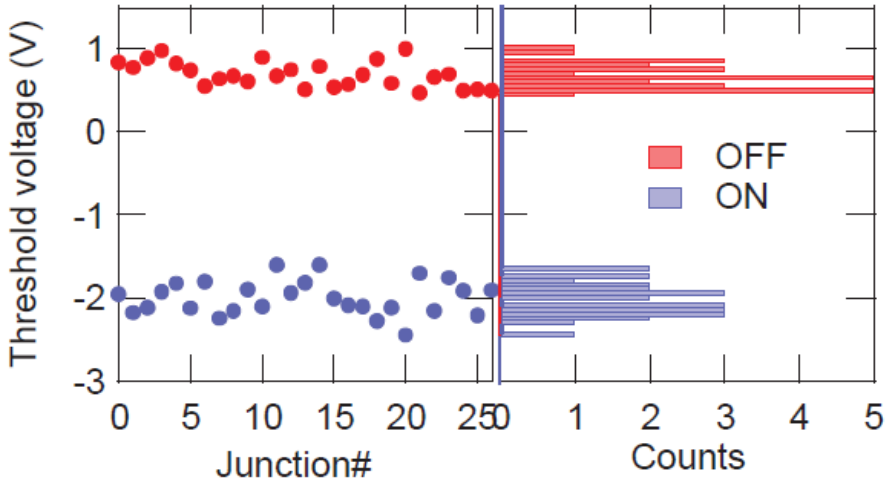
Resistive memory (memristors) 1/2

- Memristors: memories that switch due to atomic effects



Resistive memory (memristors) 2/2

- Considerable variability and impredicatbility!



Memristors
HP Labs, Borghetti et al, Nature 464 (2010)

How do we normally deal with imperfections?

- Conventional microelectronics:
  - **Design for Worst Case**

Example: setting circuit's clock frequency

- $f < \frac{1}{t_{max}}$

- t_{max} : time of the longest delay that can happen within a clock cycle, assuming all the transistors are the worst possible ones!

Problem with worst case design

- It largely wastes the potential of a technology
- Significant issue with modern CMOS…

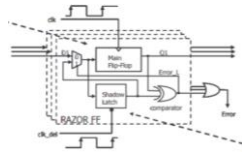
… and maybe not sustainable at all with nanodevices!!!

Two big strategies for « Better-than-worst-case » design

- Detect and correct errors
  - Example here: RAZOR
- Accept an approximate result
  - *The approach of Biology?*

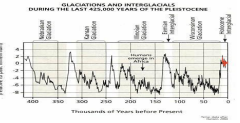
Outline of the class

- The reliability issue of nanodevices

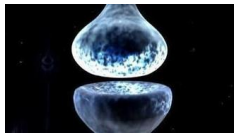


- Computing with errors: “Detect and correct”

- Approximate computation



- Computing with noise



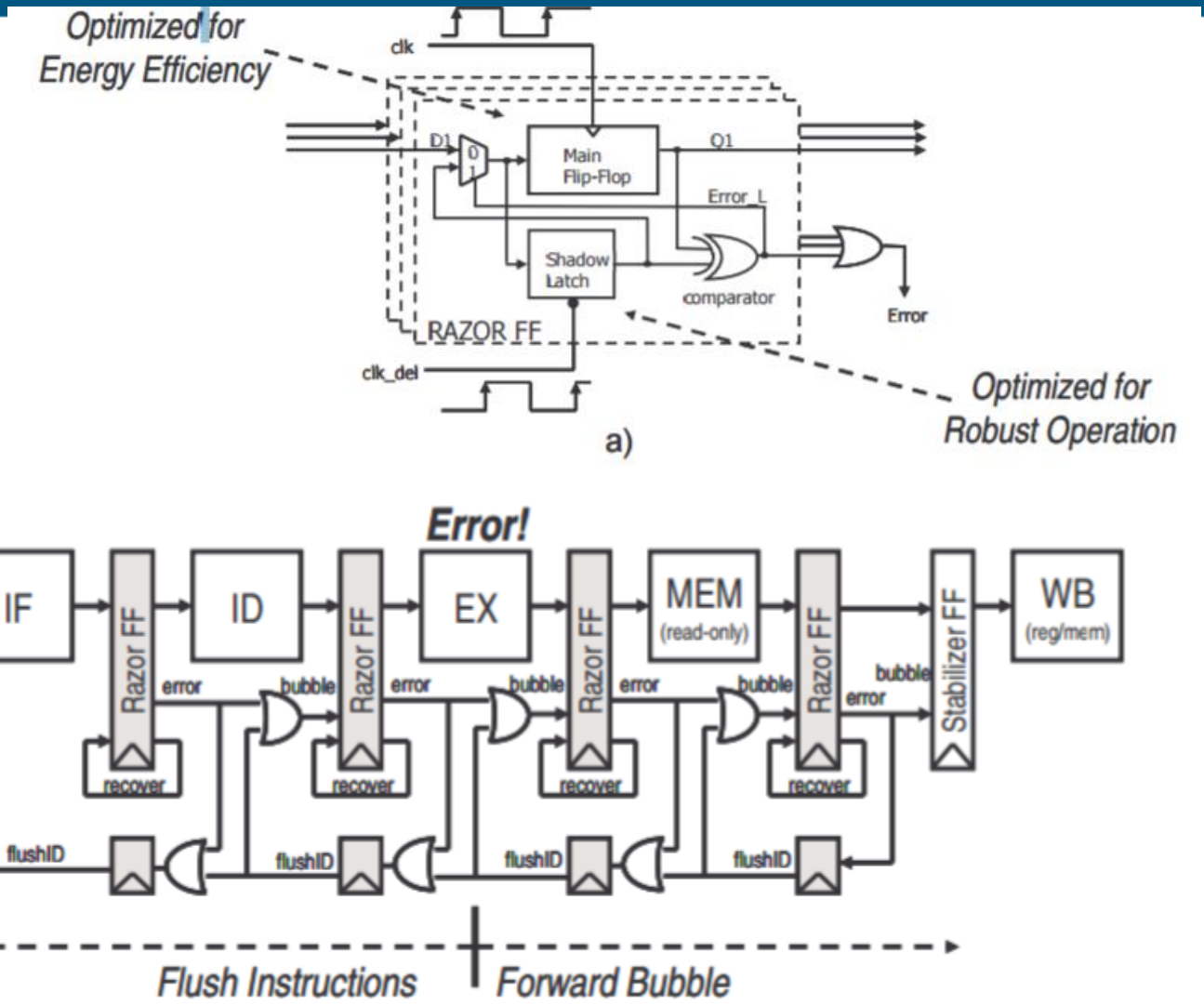
- Toward *bioinspired* computing

The RAZOR system: idea

- I want to design a low power system at *e.g.* 100MHz
- *Usually*: I choose lowest supply voltage so that circuit always works at 100MHz **in the *worst case*** *situation*
- *Here: I choose supply voltage* so that circuit ***typically*** works at 100MHz

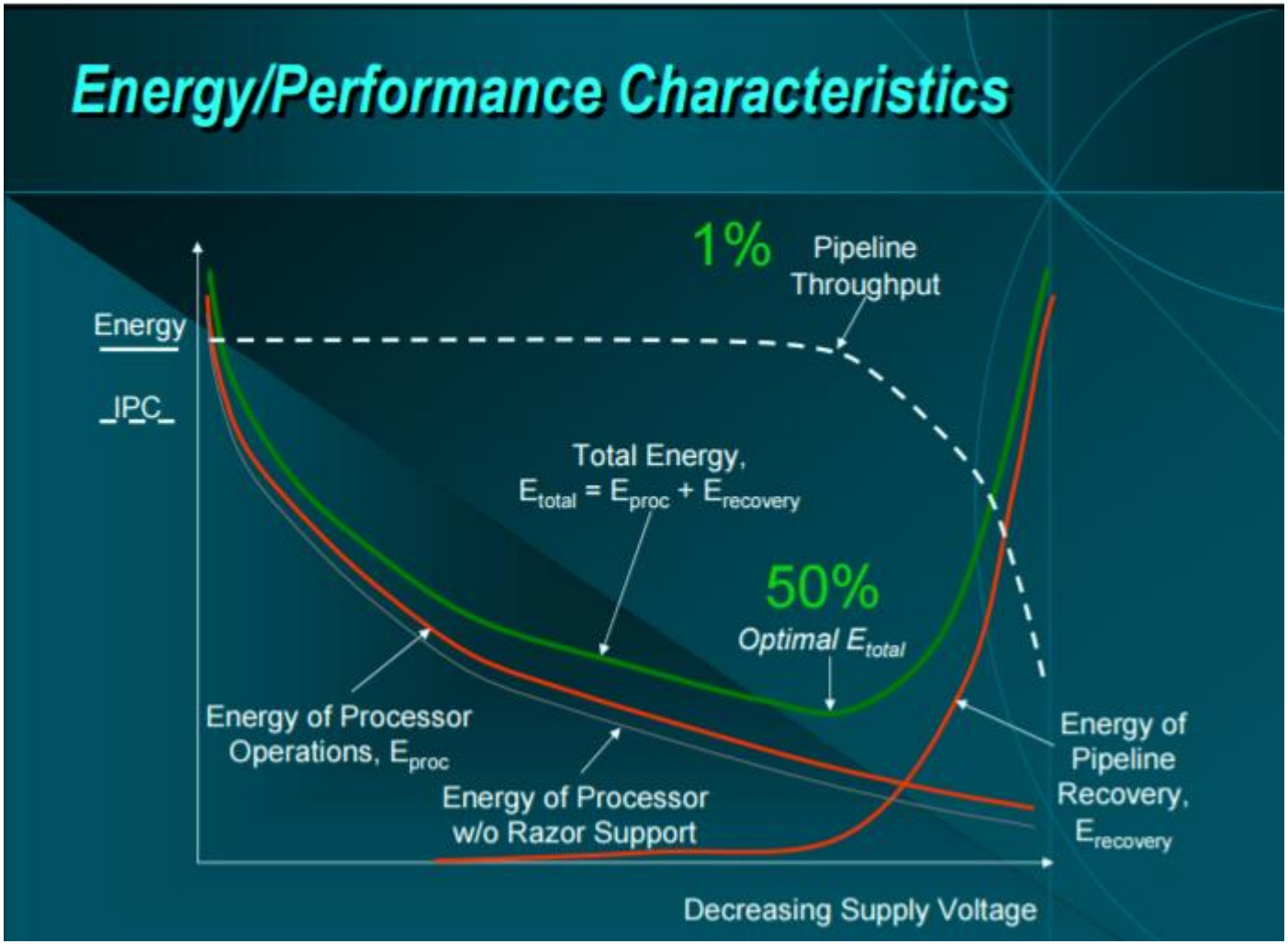
And I find a way to detect if an operation did not have time to finish so that it can be flushed

The RAZOR system



Ernst et al ASPDAC 2005

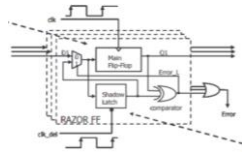
Results



Ernst et al ASPDAC 2005

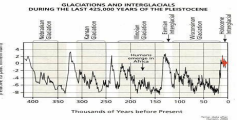
Outline of the class

- The reliability issue of nanodevices

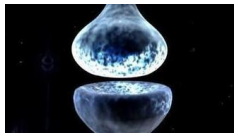


- Computing with errors: “Detect and correct”

- Approximate computation



- Computing with noise



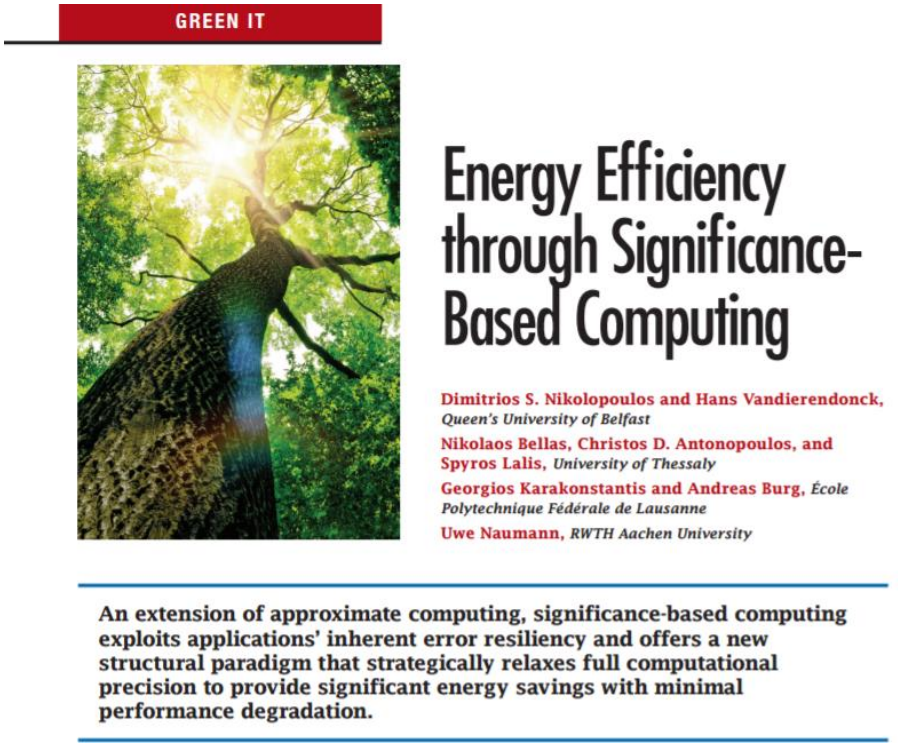
- Toward *bioinspired* computing

Approximate computing

- Let’s think about it…
- Do we *really* need absolutely exact result for everything?
- The **cost** of exactness

A current trend toward *approximate*

IEEE Spectrum 2015



Computer 2014

How are real numbers coded in electronics?

- Fixed Point representation
- Floating Point representation

Biology certainly does not use Floating Point

Modern application: Neural networks with Fixed Point

- Neural networks usually simulated with 32 bits or 64 bits floating point (GPU or CPU)

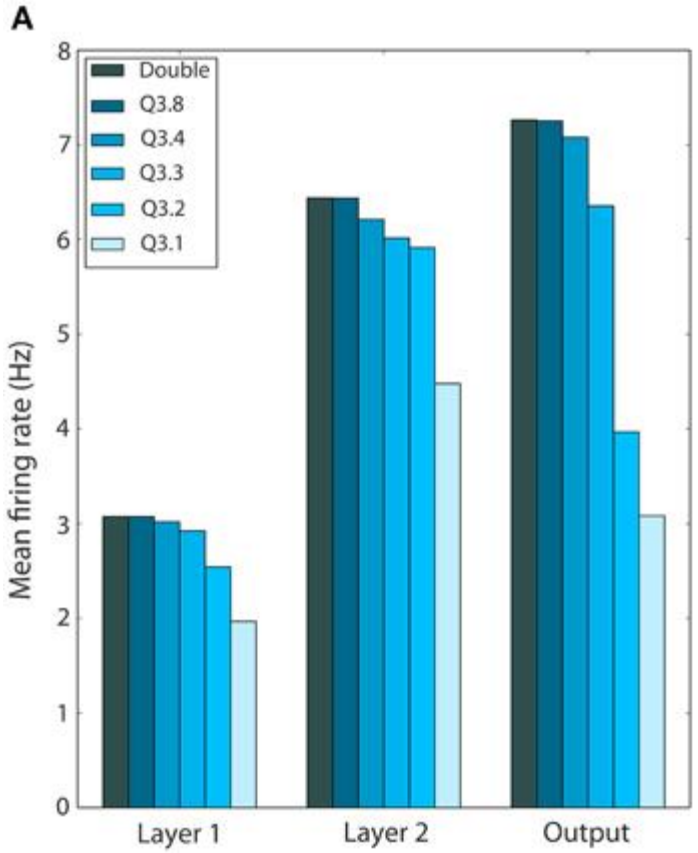
- Inference also works in low precision Fixed Point!

Approximate neural network

Table 7. CIFAR-10 classification error rate with different bit-width combinations

Activation Bit-width	Weight Bit-width 4	8	16	Float
4	8.30	7.50	7.40	7.44
8	7.58	6.95	6.95	6.78
16	7.58	6.82	6.92	6.83
Float	7.62	6.94	6.96	6.98

Lin et al, ICML 2016

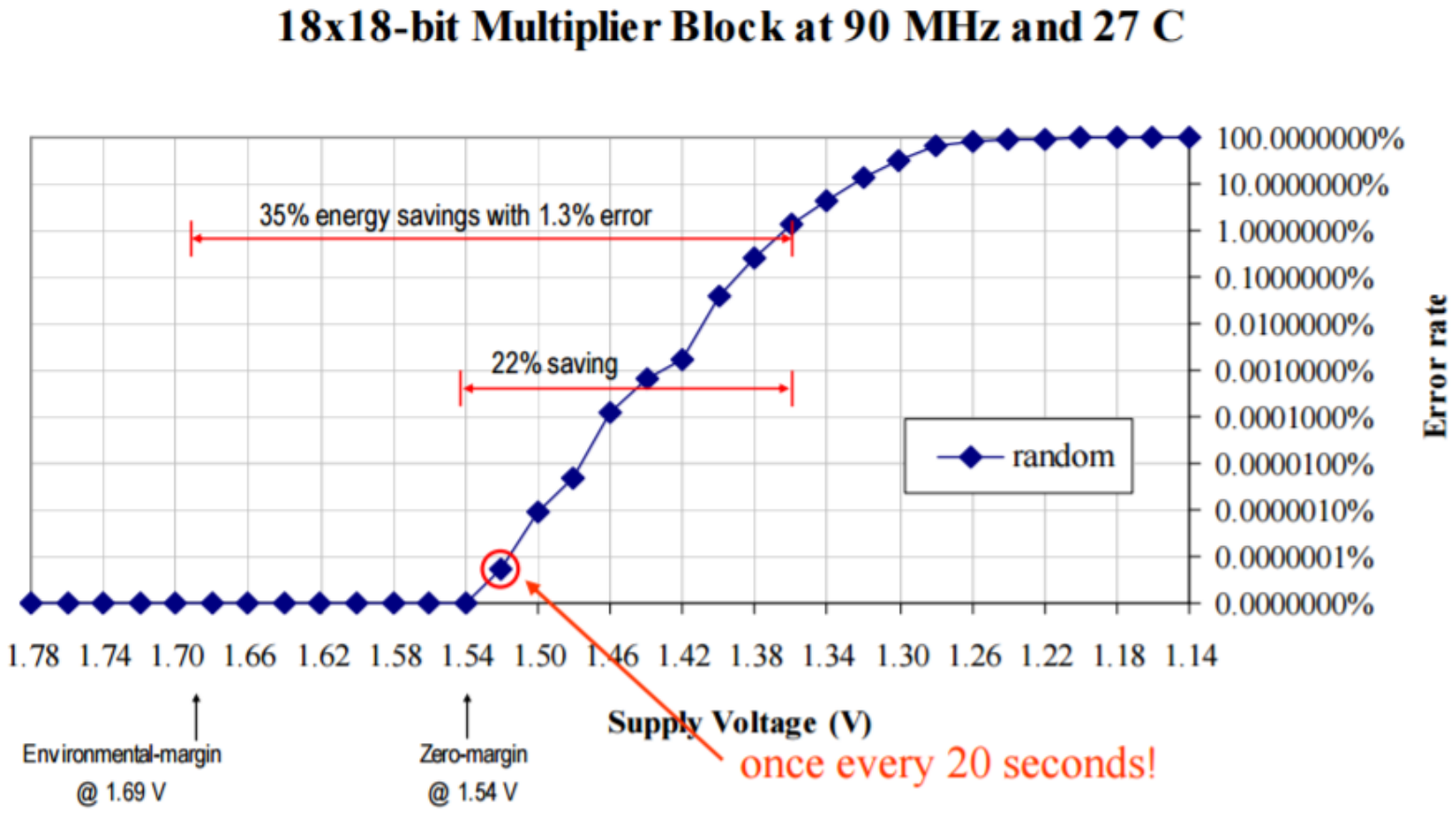


Stromatias et al, Front Neurosci 2015

Machine learning applications are especially adapted to approximate computing

Going further: Accepting incorrect least significant bits

Example with CMOS



Very significant energy saving (V^2) if you accept errors

Ernst et al ASPDAC 2005

Need for associated ecosystem

EnerJ: Approximate Data Types for Safe and General Low-Power Computation

Adrian Sampson Werner Dietl Emily Fortuna Danushen Gnanapragasam
Luis Ceze Dan Grossman

University of Washington, Department of Computer Science & Engineering
http://sampa.cs.washington.edu/

Abstract

Energy is increasingly a first-order concern in computer systems. Exploiting energy-accuracy trade-offs is an attractive choice in applications that can tolerate inaccuracies. Recent work has explored exposing this trade-off in programming models. A key challenge, though, is how to *isolate parts of the program that must be precise from those that can be approximated* so that a program functions correctly even as quality of service degrades.

We propose using type qualifiers to declare data that may be subject to approximate computation. Using these types, the system automatically maps approximate variables to low-power storage, uses low-power operations, and even applies more energy-efficient algorithms provided by the programmer. In addition, the system can statically guarantee isolation of the precise program component from the approximate component. This allows a programmer to control explicitly how information flows from approximate data to precise data. Importantly, employing static analysis eliminates

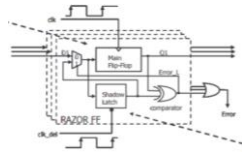
in data-centers. More fundamentally, current trends point toward a "utilization wall," in which the amount of active die area is limited by how much power can be fed to a chip.

Much of the focus in reducing energy consumption has been on low-power architectures, performance/power trade-offs, and resource management. While those techniques are effective and can be applied without software knowledge, exposing energy considerations at the programming language level can enable a whole new set of energy optimizations. This work is a step in that direction.

Recent research has begun to explore energy-accuracy trade-offs in general-purpose programs. A key observation is that systems spend a significant amount of energy guaranteeing correctness. Consequently, a system can save energy by exposing faults to the application. Many studies have shown that a variety of applications are resilient to hardware and software errors during execution [1, 8, 9, 19, 21–23, 25, 31, 35]. Importantly, these studies universally show that applications have portions that are more resilient and

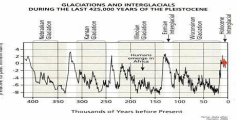
Outline of the class

- The reliability issue of nanodevices

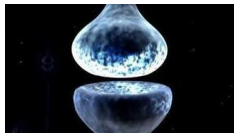


- Computing with errors: “Detect and correct”

- Approximate computation



- Computing with noise



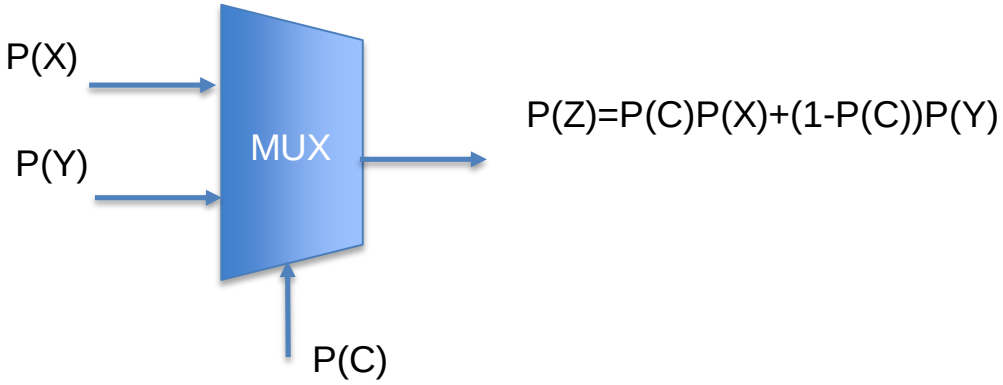
- Toward *bioinspired* computing

Can we give up determinism?

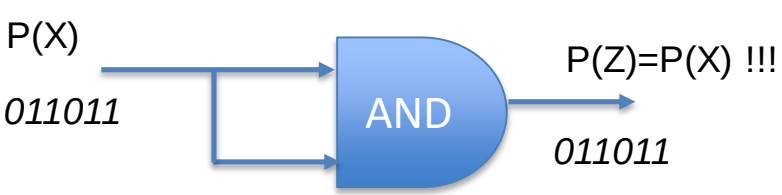
- Another idea: **stochastic computing**
- **How to compute product with stochastic bit streams? (like Gaines)**
- Precision as a function of observation time

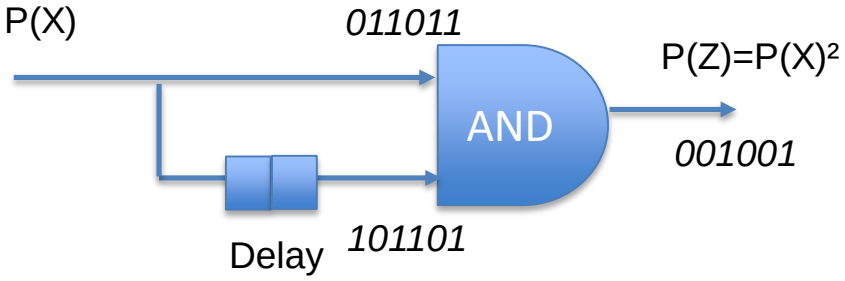
Other examples of stochastic computing

- Sum



- Square

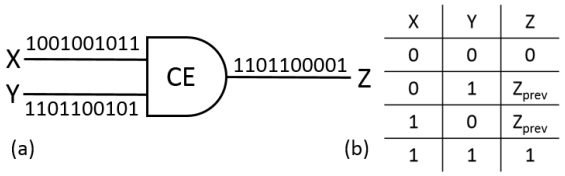




The big challenge of correlations

Stochastic computing can be very adapted to inference tasks

- **Idea: encode probabilities by probabilities**
- A single C element (8 transistors) implements Bayes rule!

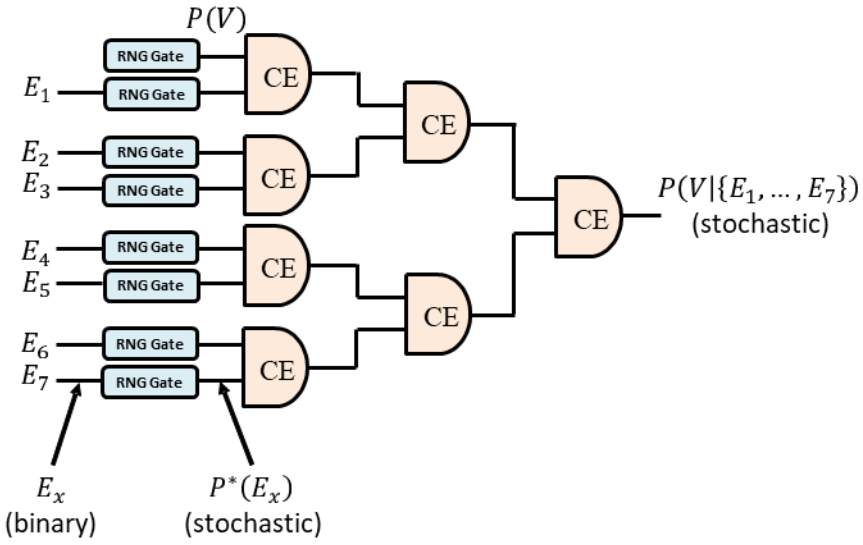


X	Y	Z
0	0	0
0	1	Z_{prev}
1	0	Z_{prev}
1	1	1

$$P(Z) = \frac{P(X)P(Y)}{P(X)P(Y) + (1 - P(X))(1 - P(Y))}$$

- They can be cascaded:

Spam detector



- But brings challenges too! (temporal correlations)

Friedman et al, IEEE TCAS I, 2016; IJAR 2017

Example of spam classification

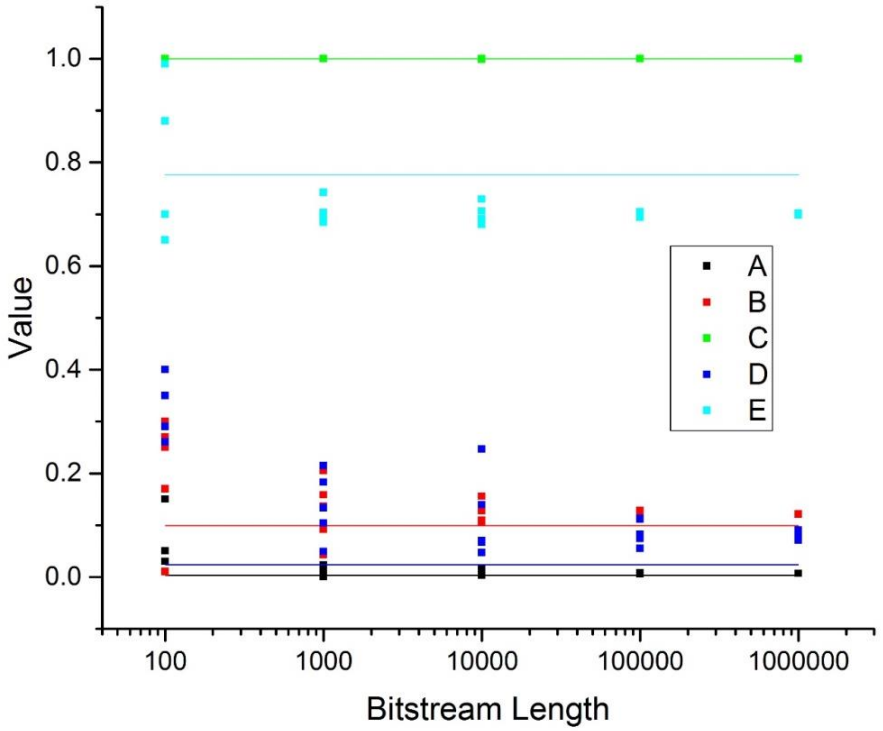
A Do you want to get pizza for lunch?

B Can you please check my stochastic simulations?

C If you want to earn a fortune, send a $100 check to Nigeria and we will transfer $10,000 to your account.

D My weekly commute to Nigeria includes a transfer in Morocco. I will check if my flight is on schedule- if so, do you want to get pizza when I arrive?

E There is a $10 fee for all check transfers.

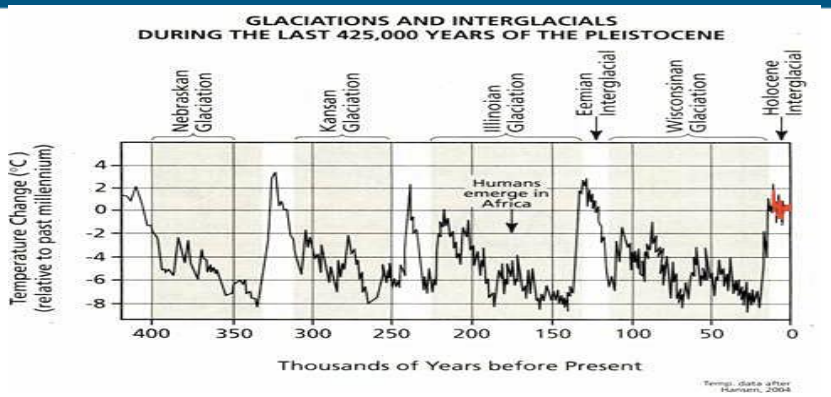


A promising lead, lots of work necessary

Friedman et al, IEEE TCAS I, 2016; IJAR 2017

Crazier ideas: stochastic resonance

First understood in the 80's for quaternary glaciations [C.Nicholis]
Now also observed in physics and biology

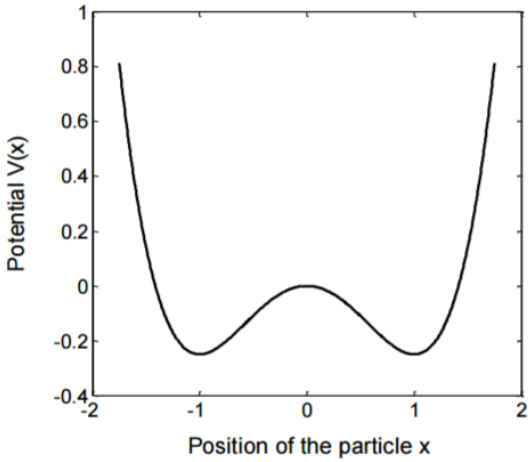


Idea: increasing a system's sensitivity to small inputs by **adding** noise

- Seen in neurons in vitro
- Some evidence suggests it may be used by the brain

Similarly, some circuits can have behavior improved by noise!

The canonical theory of stochatic resonance



A hypothetical particle that evolves along

$$\dot{x}(t) = -V'(x) + A_0 \cos(2\pi F t) + \chi(t)$$

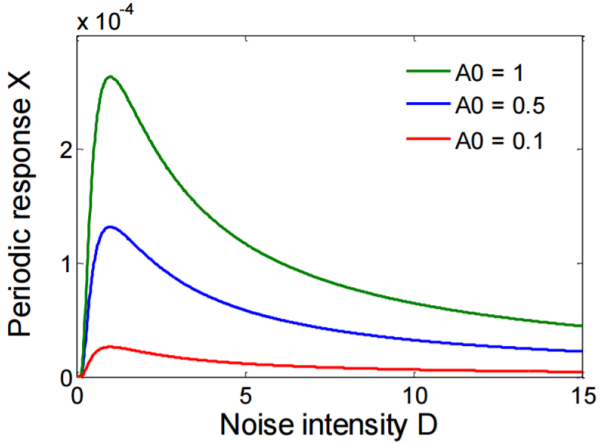
Bistability **Subthreshold drive** *Noise (D)*

Periodical part of the response

$$X(D)\ \cos(2\pi F t + \Phi(D))$$

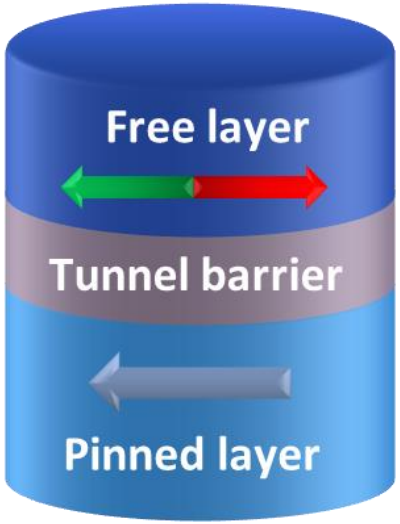
$$X(D) = \frac{A_0}{D} \frac{2 r_K(D)}{\sqrt{4 r_K(D)^2 + 4\pi^2 F^2}}$$

$$r_K = \frac{1}{\sqrt{2}\pi} \exp\left(-\frac{\Delta V}{D}\right)$$



Periodical response has a maximum as a function of noise

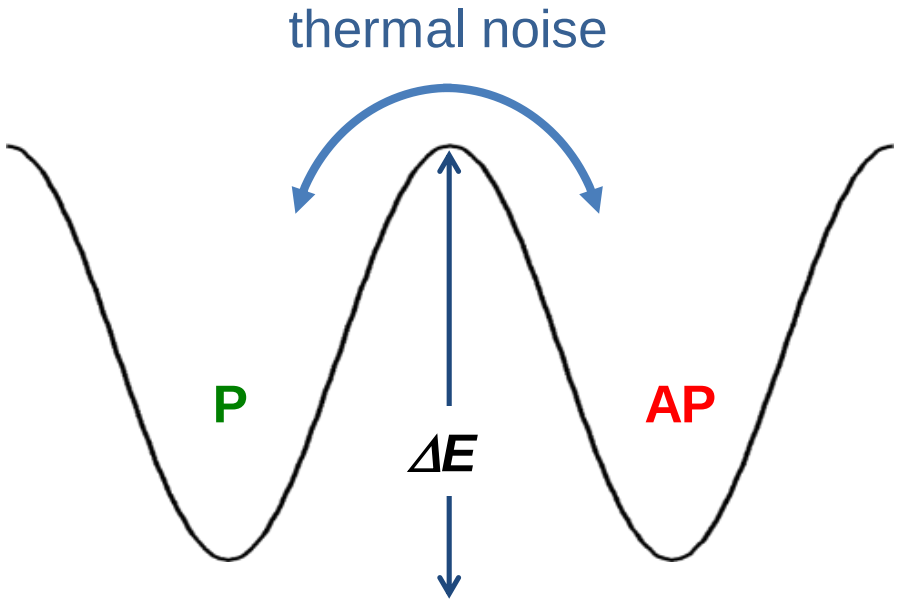
Superparamagnetic Tunnel Junction



CoFeB 1.7 nm

MgO 1.0 nm

CoFe 2.5 nm
Ru 0.85 nm
CoFeB 2.4 nm

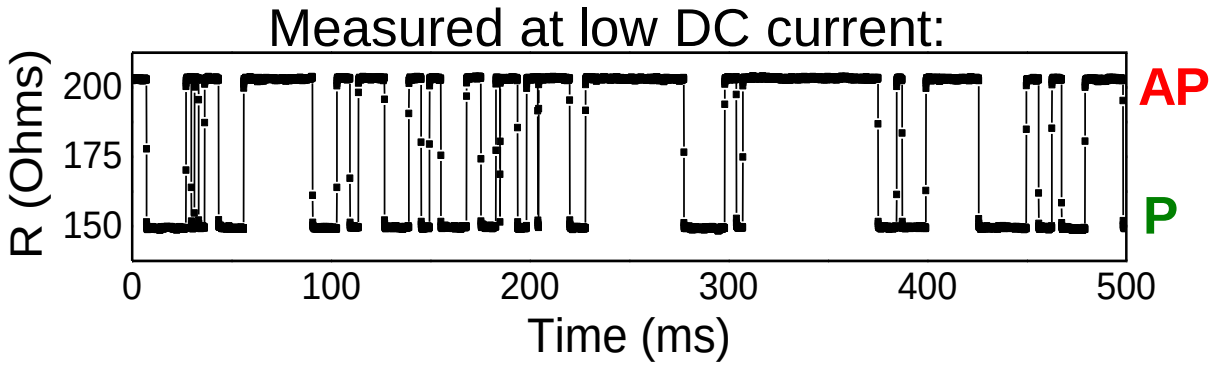


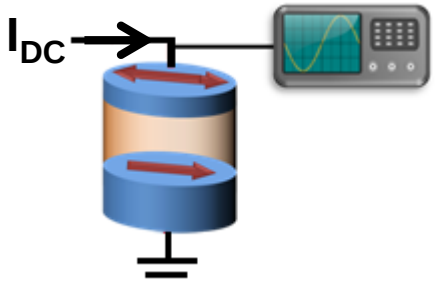
Same basic technology as Spin Torque MRAM

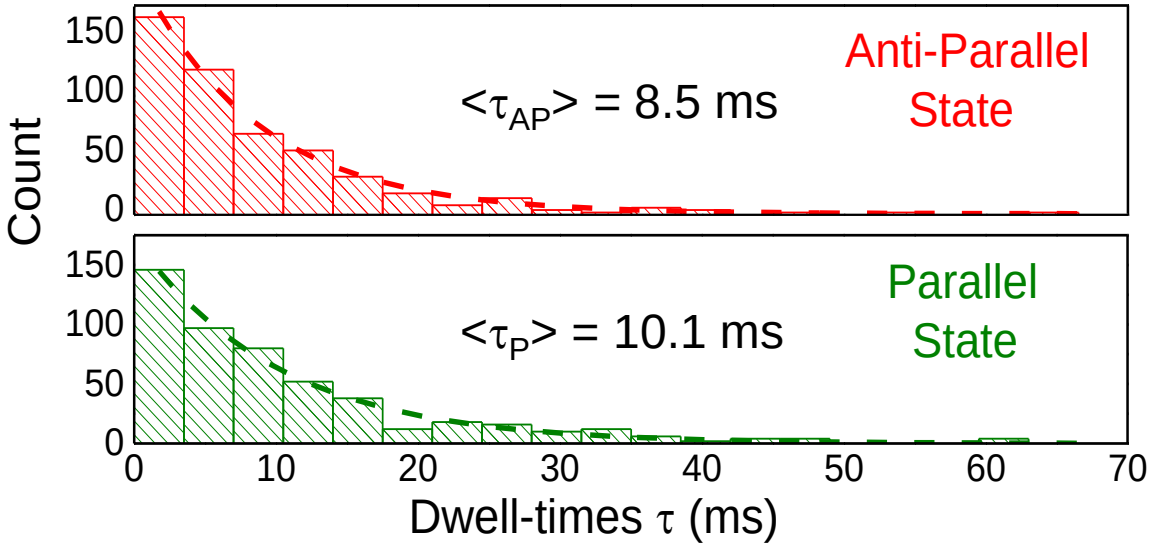
Energy barrier ΔE depends on material and volume of the free layer

Small size → low barrier → easy switching
→ high frequency oscillations

Poissonian Statistics of Switching







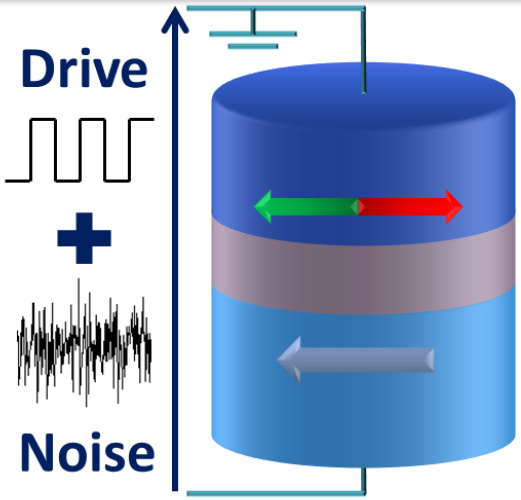
Poisson process

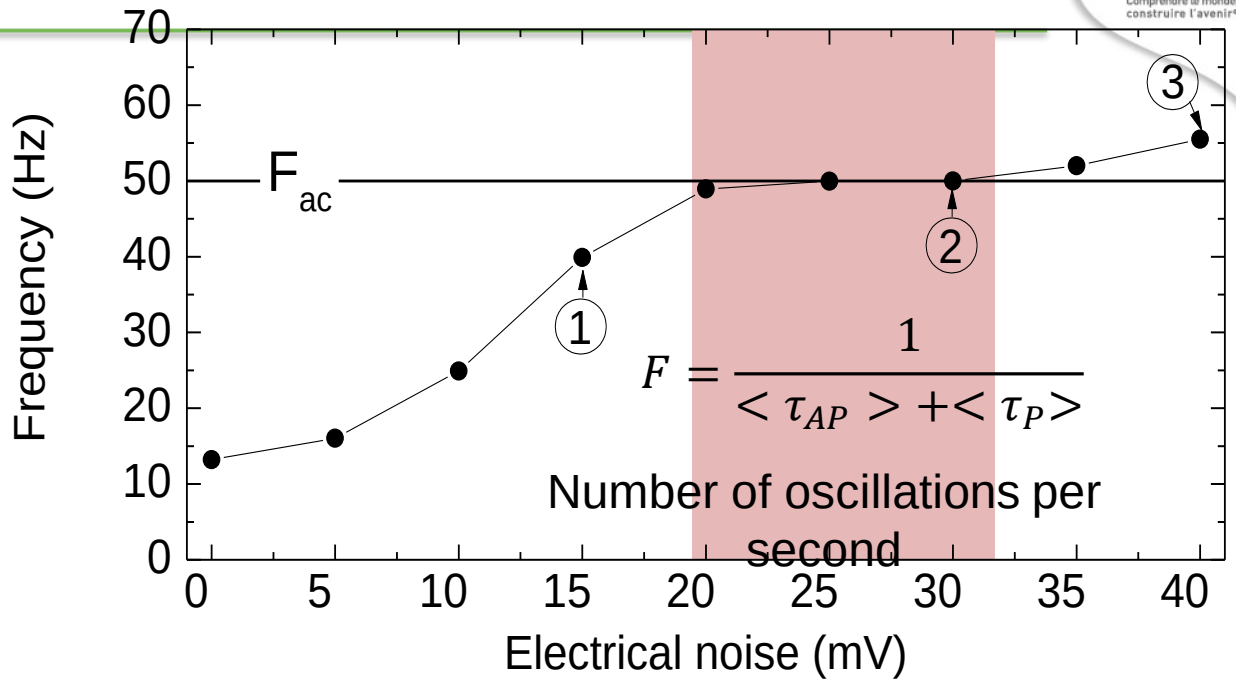
$$F_0 = \frac{1}{<\tau_P>+<\tau_{AP}>} = 53.8\ Hz$$

F_0 : natural frequency with no source

Locatelli et al, Phys Rev Appl 2014; Mizrahi et al, IEEE Trans Mag 2015

Experiment: Noise Controls Frequency and Phase Locking

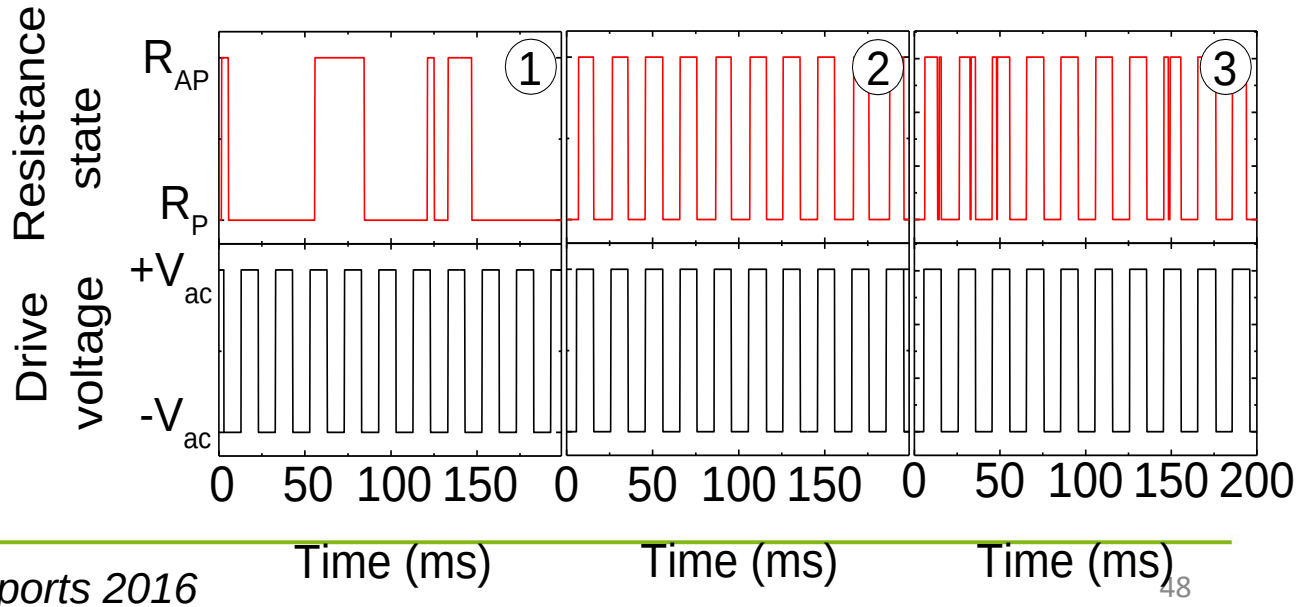




V_{ac} = 63 mV while
V_c = 235 mV @0K

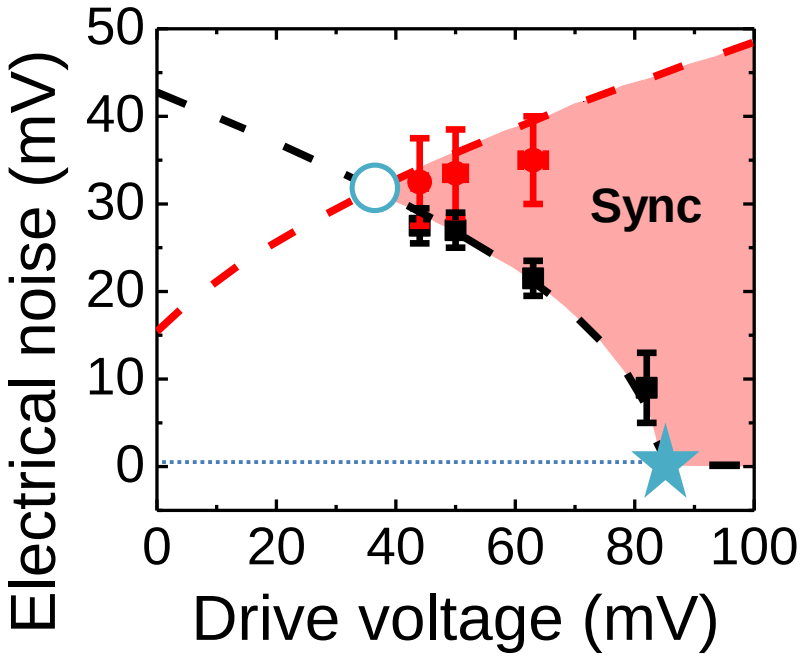
$\Delta E = 22.5\ k_B T$
Natural frequency ≈ 0.1 Hz

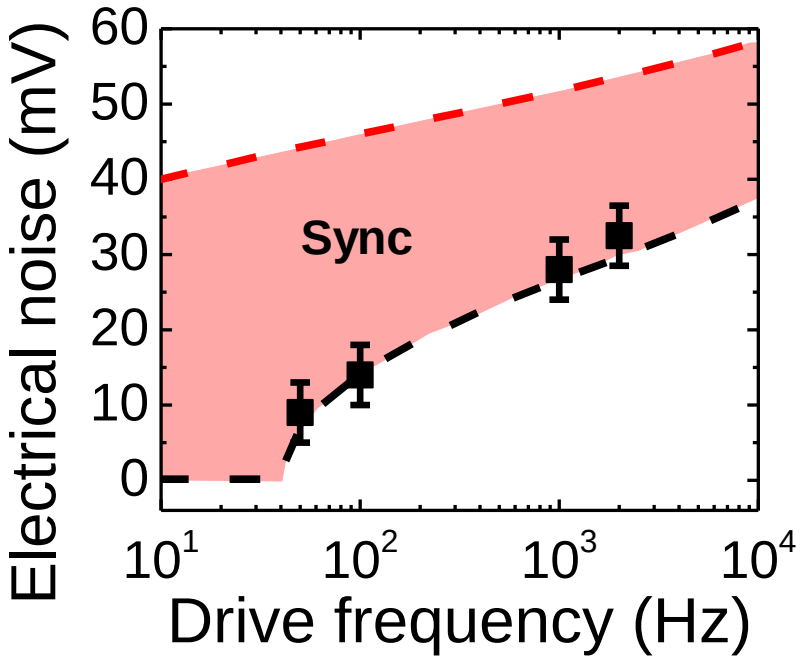
Thermal noise (room temperature)
+
White electrical noise



Mizrahi et al., Scientific Reports 2016

Effect of Drive Amplitude and Frequency





Boundaries synchronization

Model

Experiments

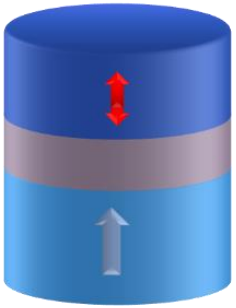
Natural frequency ≈ 0.1Hz

→ **Synchronization possible at broad ranges of amplitudes and frequencies**

Mizrahi et al., Scientific Reports 2016

Minimum Energy Required for Synchronization

Out of plane magnetic tunnel junction

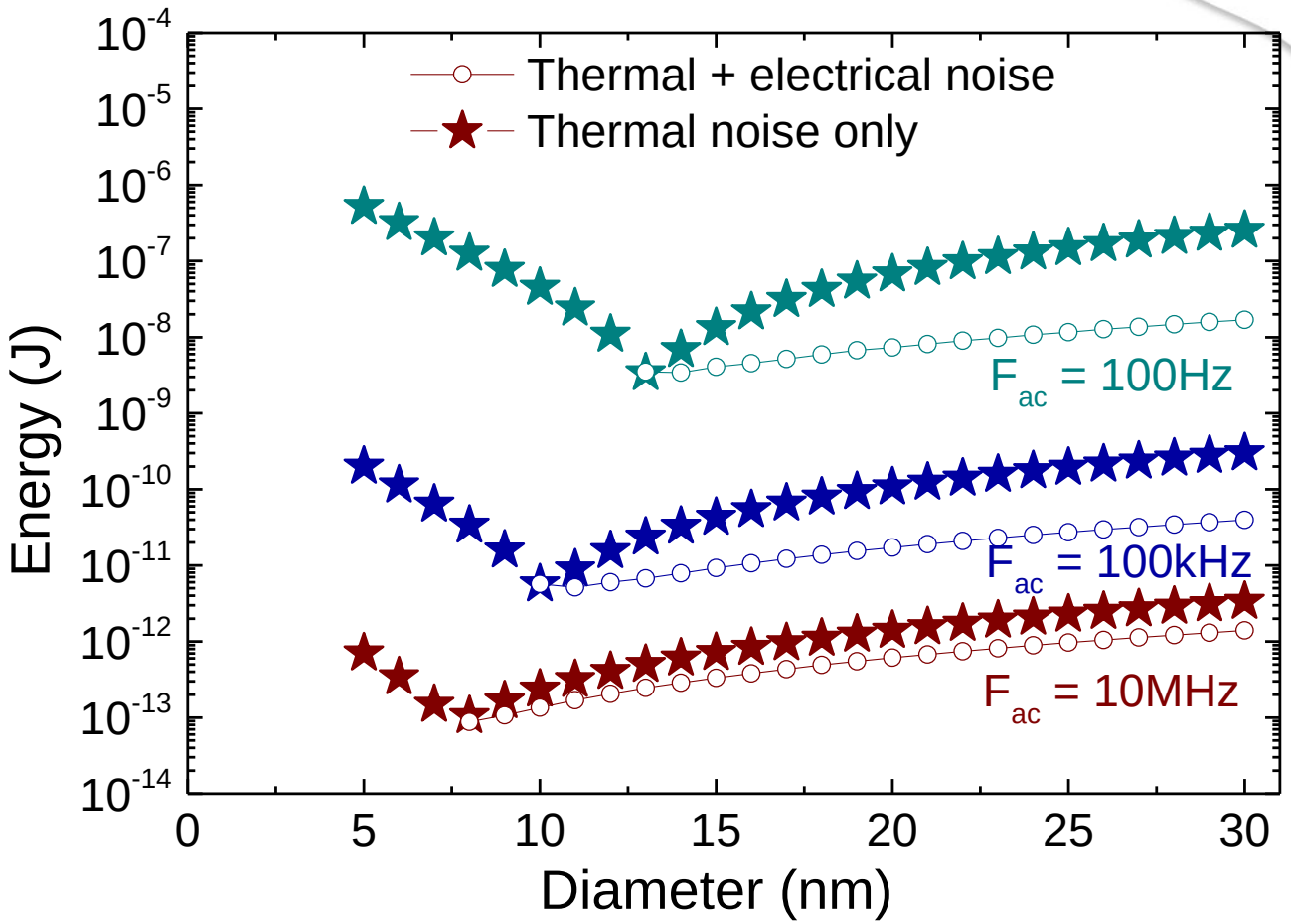


$$\Delta E = \Delta E_0 \frac{D^2}{{D_0}^2}$$

$$R = \frac{RA}{\pi \frac{D^2}{4}}$$

$$V_c = cte$$

Sato et al., APL, 2014



- **Optimal junction diameter (natural frequency ≈ drive frequency)**
- **Adding electrical noise is energy efficient**
- **$E_{min} < 10^{-13}$ J**

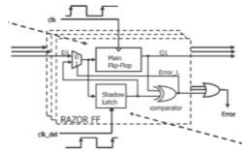
Mizrahi et al., Scientific Reports 2016

Industrial challenge of Better than Worst Case

- How does industry feel about better than worst case design?

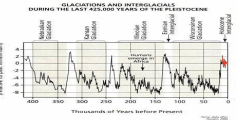
Outline of the class

- The reliability issue of nanodevices

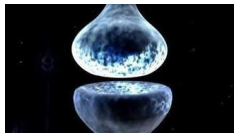


- Computing with errors: “Detect and correct”

- Approximate computation



- Computing with noise



- Toward *bioinspired* computing

Neuroinspiration

- Biology does not work with digital logic, uses approximate and redundant coding, can perform advanced computation, at low power

- *A supercomputer (MW) cannot do what a brain does (20W)*

vs.

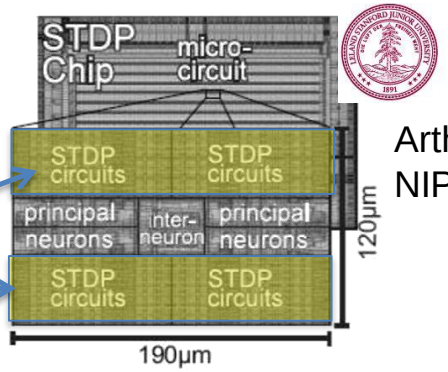
Main ideas

- Nanodevices with a lot of **functionality**
- Massive **parallelism**
- **Slow, low power operations**

An example: Synaptic computation

- Synapse = self-adapting connection between two basic computing units *(can change electrical resistance)*

- Synapse = *transmission + learning*



Arthur, NIPS 2006

With CMOS, difficult to cointegrate logic/memory
But natural with nanoelectronics

Bioinspired nanoelectronics

- Some nanodevices encompass memory (e.g. spin torque magnetic memory)
- Perfect element for implementing synapses!
- Neural networks have excellent resilience to their elements' imperfections

BIOCOMP: G. W. Burr's lecture

Conclusions

- Nanodevices tend to be less reliable than traditional electron devices
- Worst case design may not be sustainable with nanodevices
- Alternate computing approaches that detect errors or accept approximate results may better benefit from nanodevices qualities
- Groundbreaking ideas (computing with noise or like biology) may allow true reinvention of computing with nanodevices

Thank you for your attention!