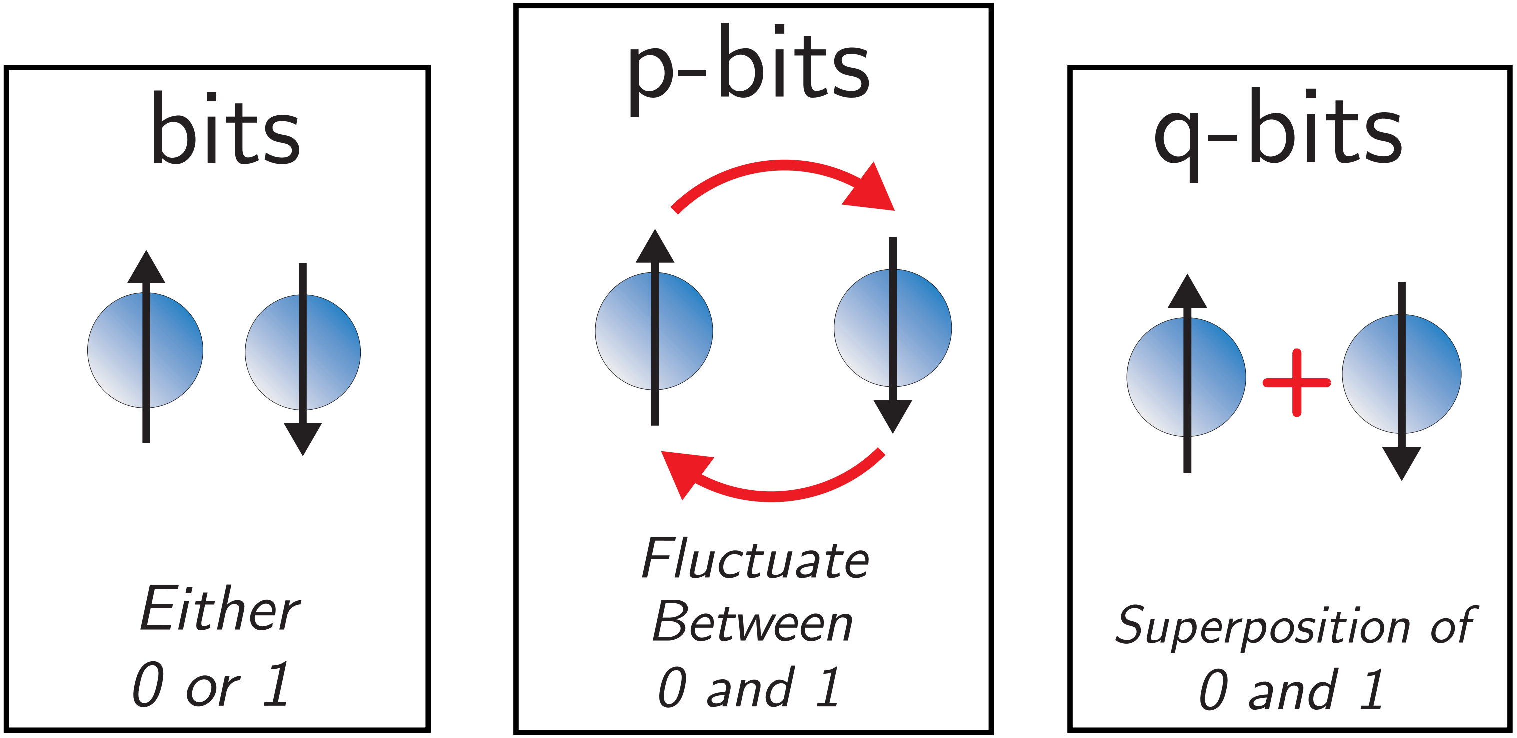


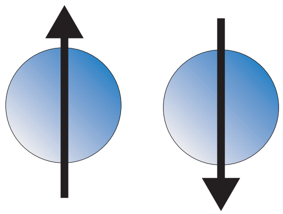
Probabilistic Computing: From materials and devices to circuits and systems

Kerem Camsari
Postdoctoral researcher

03/04/2020

Computing today...

bits



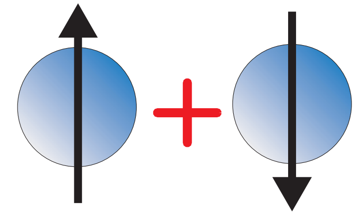
Either 0 or 1

Domain Specific Hardware

Machine Learning

Quantum Computing

q-bits

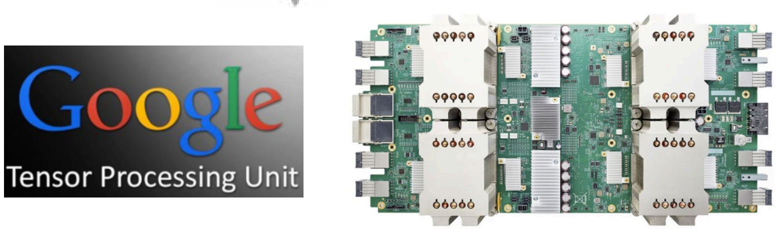


Superposition of 0 and 1

CPU

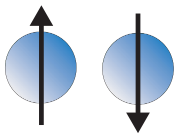
GPU

TPU



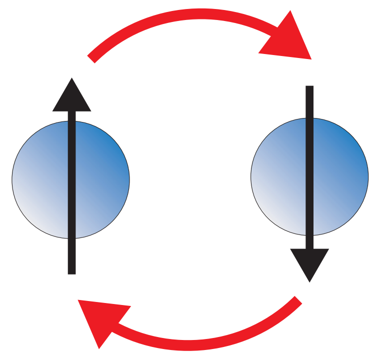
Probabilistic or p-bits

bits



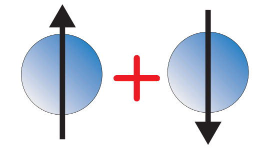
Either
0 or 1

p-bits

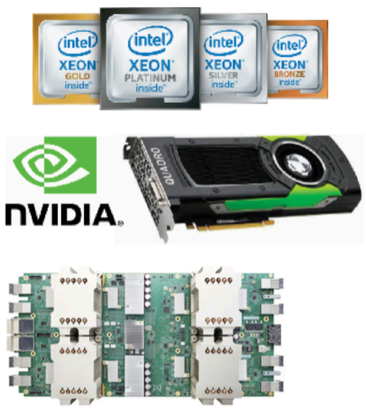


Fluctuate
Between
0 and 1

q-bits



Superposition of
0 and 1



Domain specific

Machine Learning

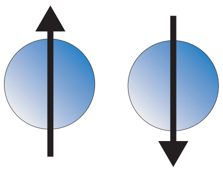
NISQ-era QC

(Noisy-Intermediate Scale Quantum)

Materials/Devices/Circuits/Systems/Algorithms

bits, p-bits, q-bits

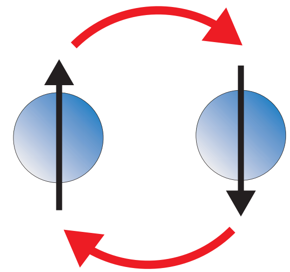
bits



Either
0 or 1

Naturally deterministic

p-bits



Fluctuate
Between
0 and 1

Naturally stochastic

q-bits

$f(N)$:
$\exp(N) \rightarrow \log(N)$
Shor's algorithm

Naturally quantum mechanical

delay ← $\tau = \tau_0 \, f(N)$ → problem, algorithm dependent

energy ← $E = E_0 \, g(N)$ → problem, algorithm dependent

τ_0, E_0 → device, architecture dependent

N: problem size

Moore's Law: All about reducing $E_0 \tau_0$

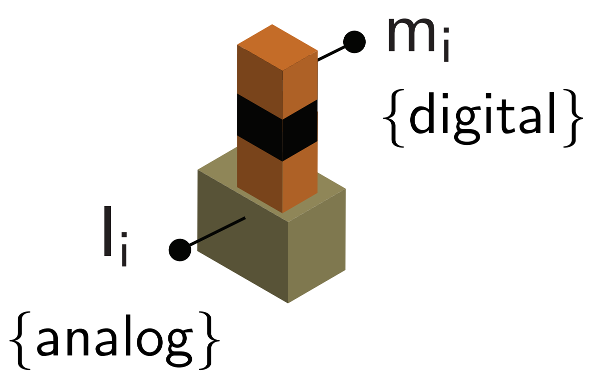
Outline

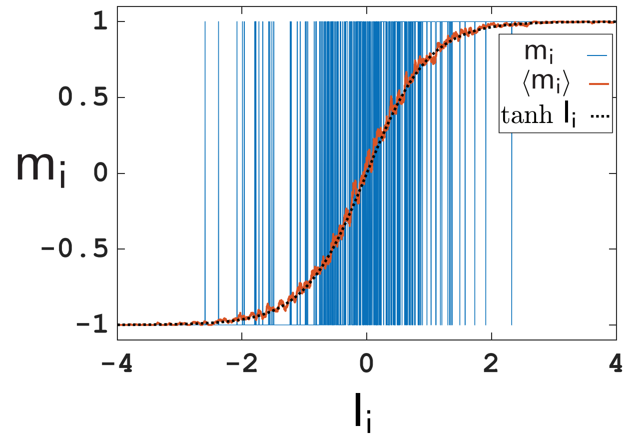
- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit
- Future directions

Outline

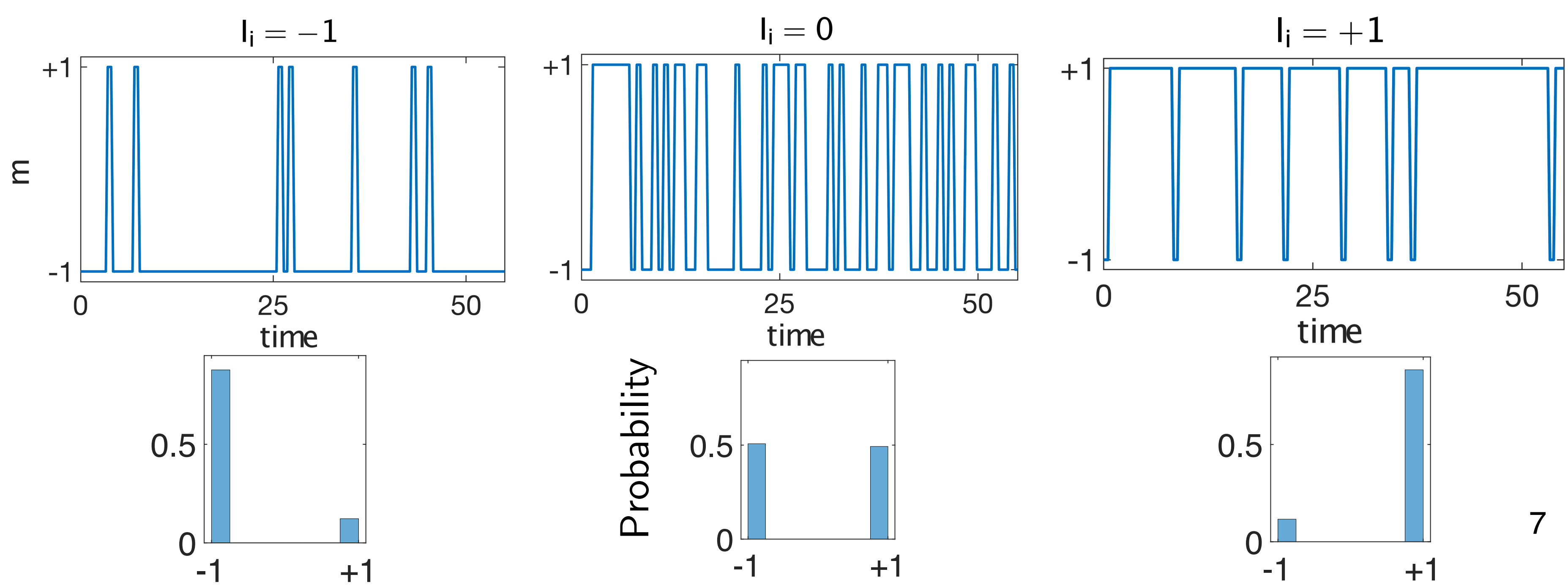
- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit
- Future directions

What is a p-bit?

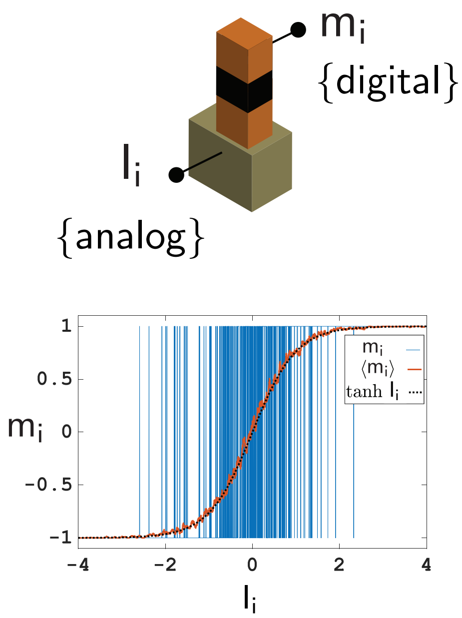


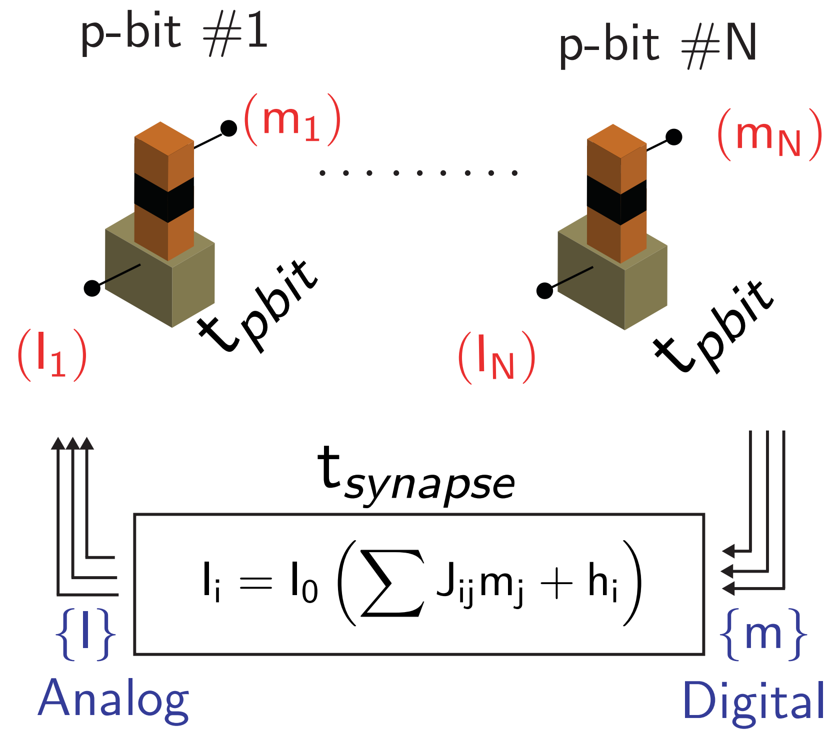


$$m_i = \text{sgn}\left[\left(\tanh I_i - \text{rand}(-1, 1)\right)\right]$$



Architecture of a p-computer





- Clockless operation

- Massively parallel: Performance increases with increasing # of p-bits.

- Requires a fast synapse!

$m \rightarrow I \qquad t_{synapse}$

$I \rightarrow m \qquad t_{pbit}$

$$t_{synapse} \lesssim t_{pbit}$$

A wide application space for p-computers

Quantum Computing inspired

Invertible Logic

Adder ⟷ **Subtractor**

(1) Subset Sum Problem

Multiplier ⟷ **Divider**

(2) Prime Factorization

Camsari et al., ***Phys. Rev. X, (2017)***

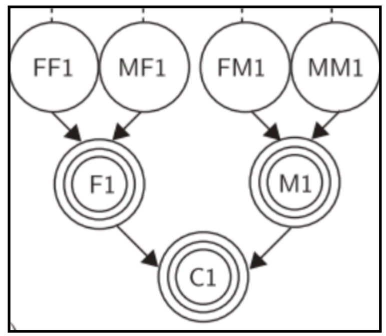
Combinatorial Optimization (e.g. TSP)

$$\underbrace{\arg\min}_{\mathbf{m} \in \text{phase space}} \underbrace{E}_{\text{cost function}} (\mathbf{m})$$

Sutton et al., ***Sci. Rep., (2017)***

Machine Learning inspired

Bayesian Networks



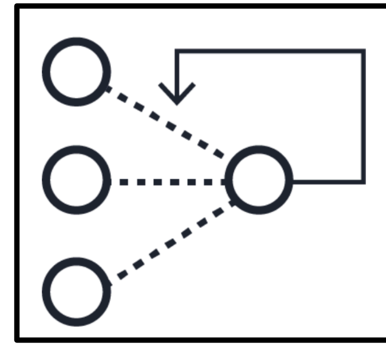
in-situ learning

$\langle m_i m_j \rangle$

inference

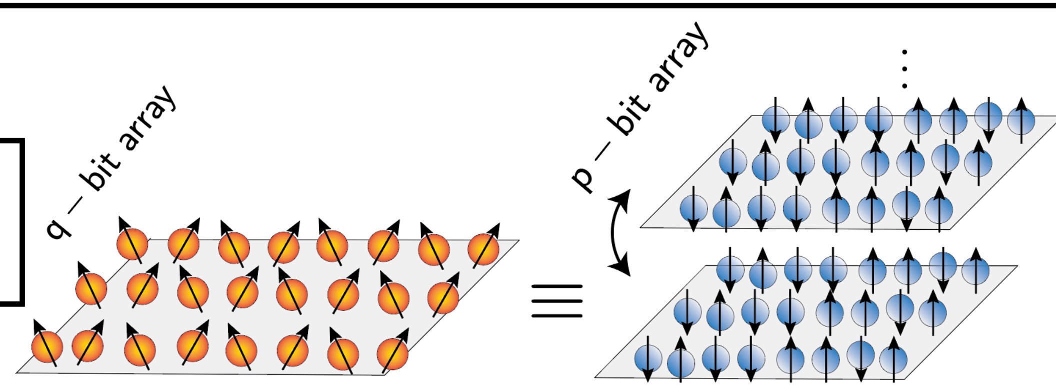
Faria et al., ***AIP Advances (2018)***

Boltzmann Machine



Kaiser et al., ***Frontiers in Comp. Neuroscience (2020)***

Quantum systems emulated with p-bits



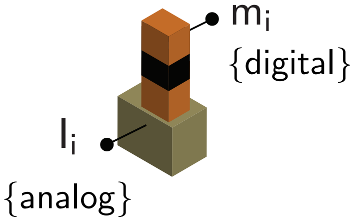
Camsari et al., ***Physical Review Applied (2019)***

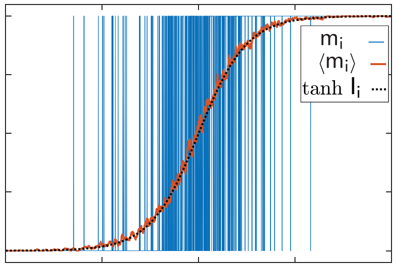
Outline

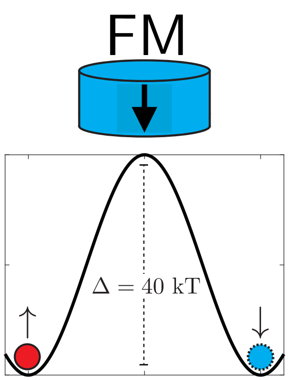
- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit

- Future directions

How to build a p-bit?

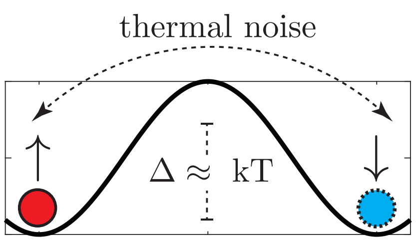






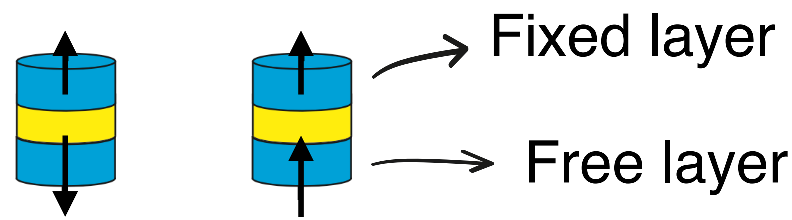
$$\tau = \tau_0 \exp(\Delta/kT)$$
$$\tau_0 \approx \text{ps} - \text{ns}$$

$$\Delta = 40 - 60 \text{ kT}$$
$$\tau \approx \text{ years}$$

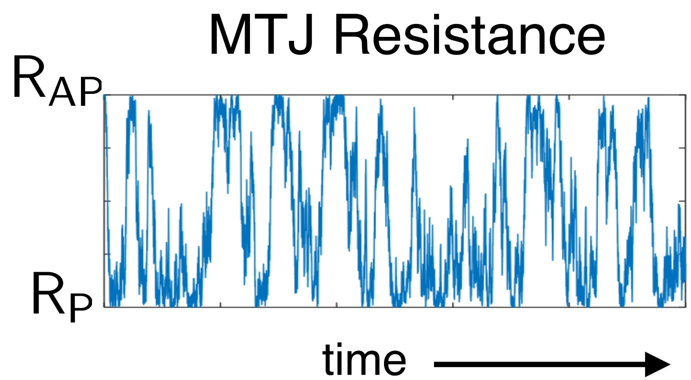


$$\Delta = kT$$
$$\tau \approx \text{ps} - \text{ns}$$

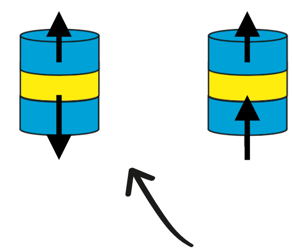
Magnetic Tunnel Junction (MTJ)



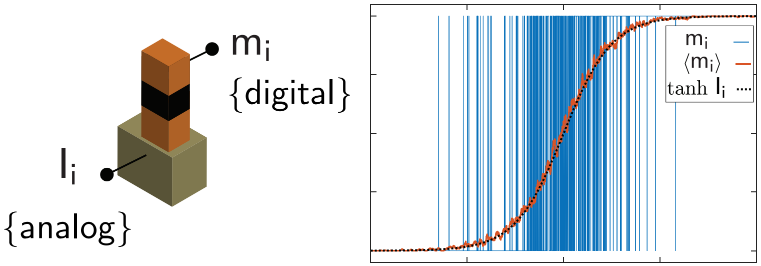
$$R_{AP} > R_P \qquad R_{AP}/R_P \approx 2 - 3$$

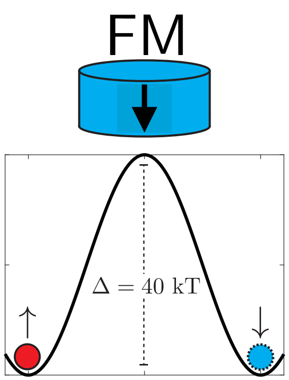


(ps – ns) depending on magnet design



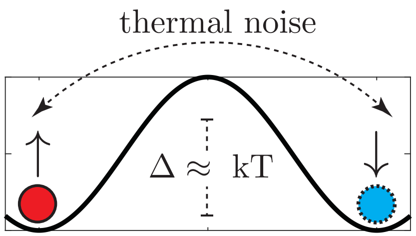
How to build a p-bit?



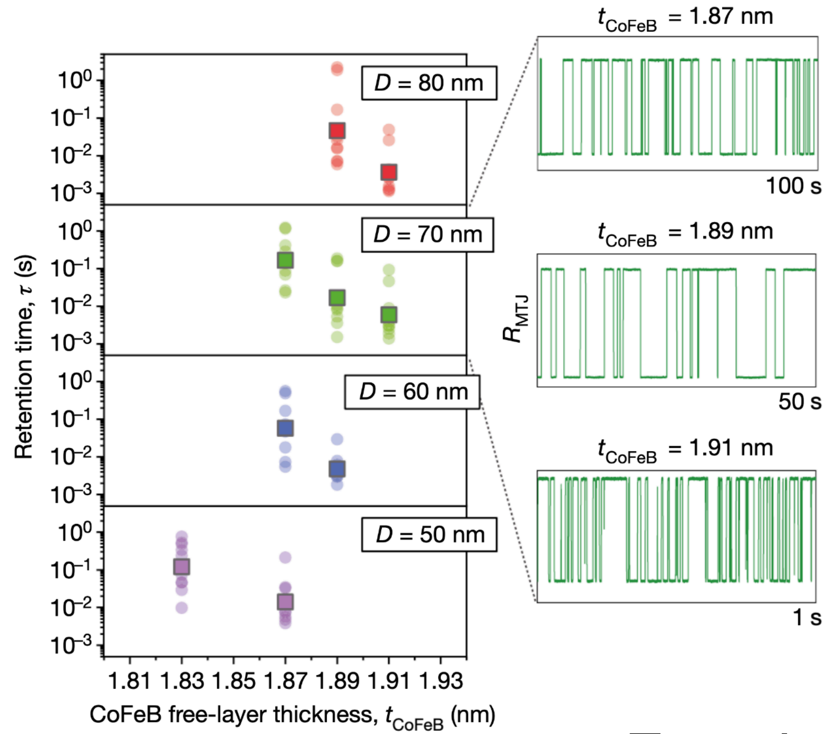


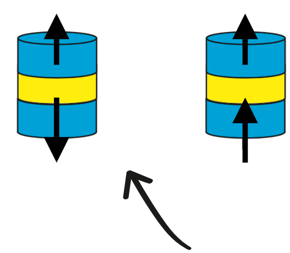
$$\tau = \tau_0 \exp(\Delta/kT)$$
$$\tau_0 \approx \text{ps} - \text{ns}$$

$$\Delta = 40 - 60 \text{ kT}$$
$$\tau \approx \text{ years}$$



$$\Delta = \text{kT}$$
$$\tau \approx \text{ps} - \text{ns}$$





Unstable

free layer

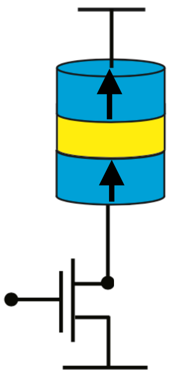
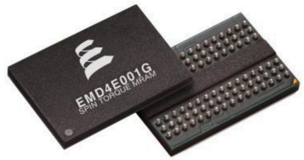
Experimental

PMA-MTJs

(Tohoku

University)

Embedded MTJ based p-bit

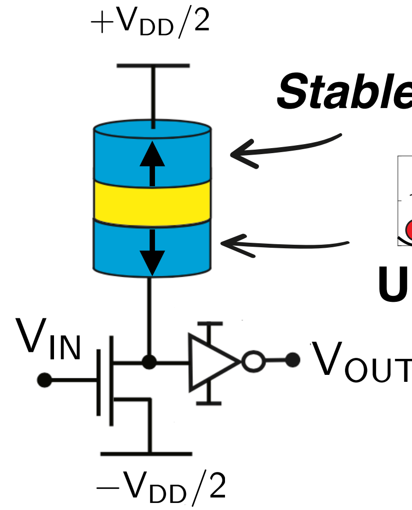



STT-MRAM

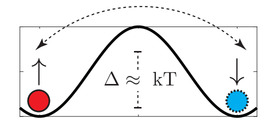
1Gb MRAM

1T/1MTJ

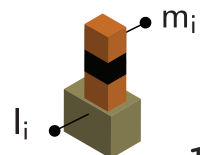
-Everspin
-Samsung
-GlobalFoundries



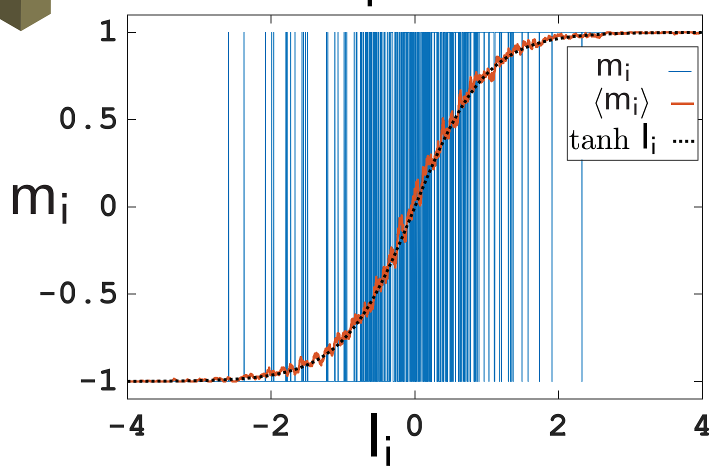
Stable fixed layer



Unstable free layer

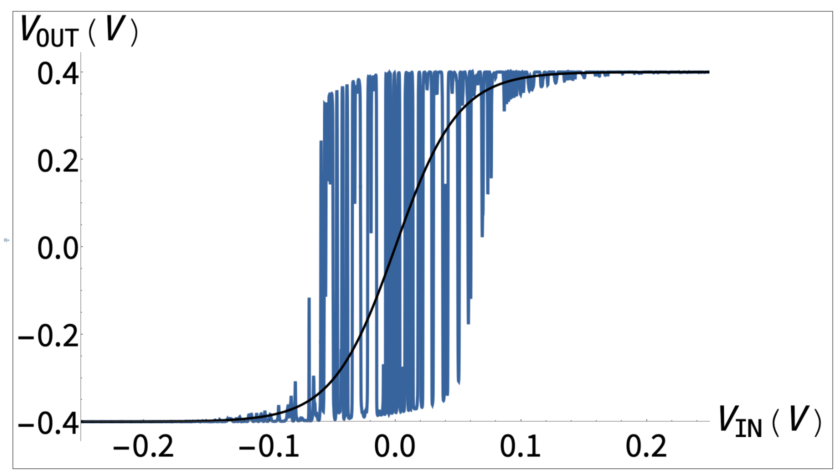


Ideal: Behavioral equations



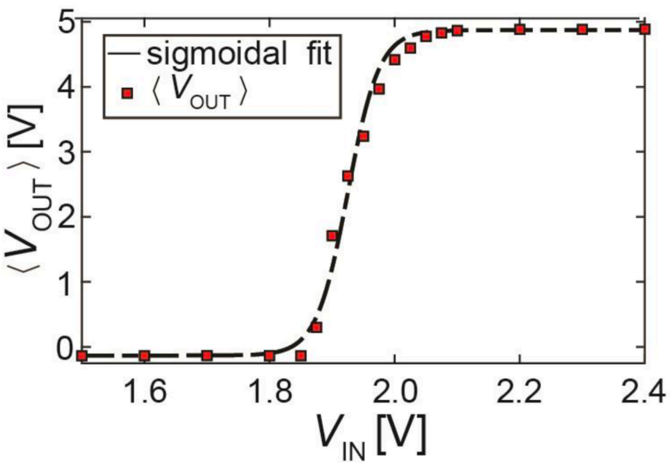
Camsari et al., *Phys. Rev. X (2017)*

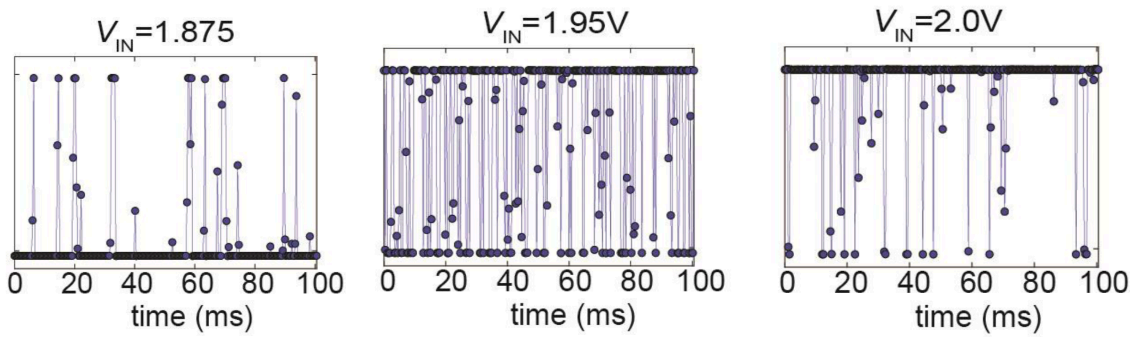
SPICE: PTM/BSIM Models + Stochastic LLG

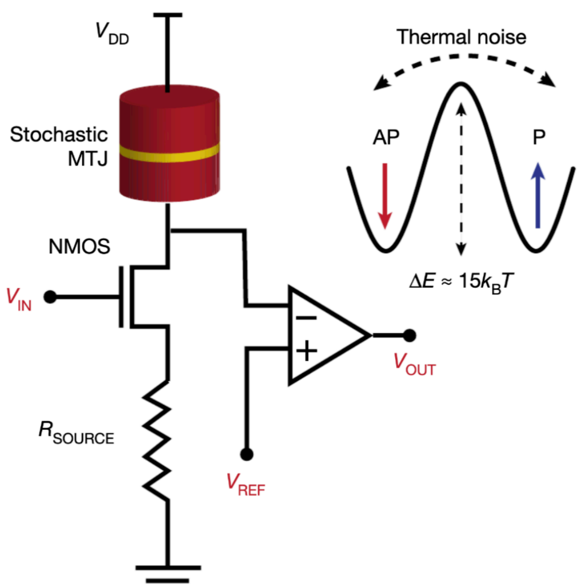


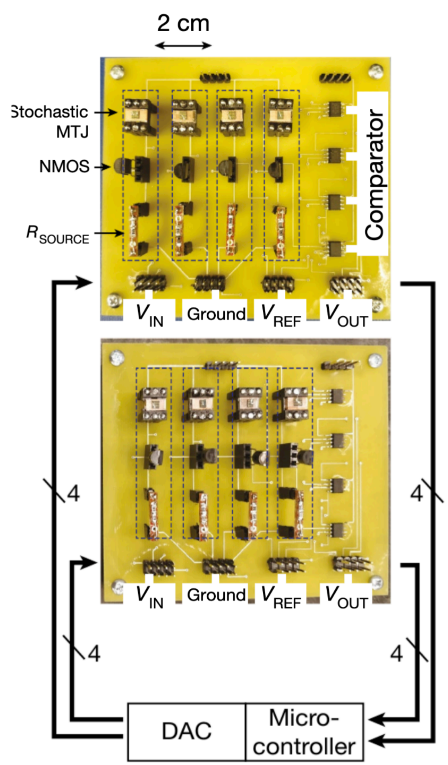
Camsari et al., *IEEE EDL (2017)*

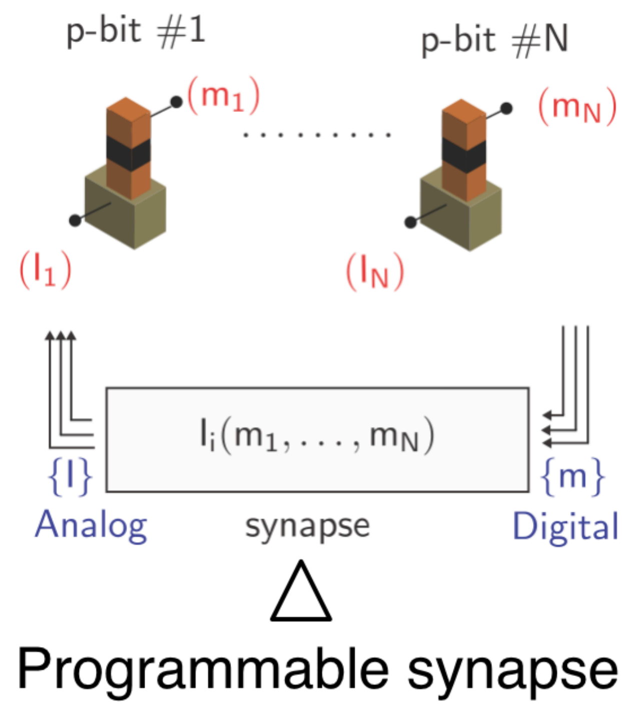
Prototype p-computer (8 p-bits)







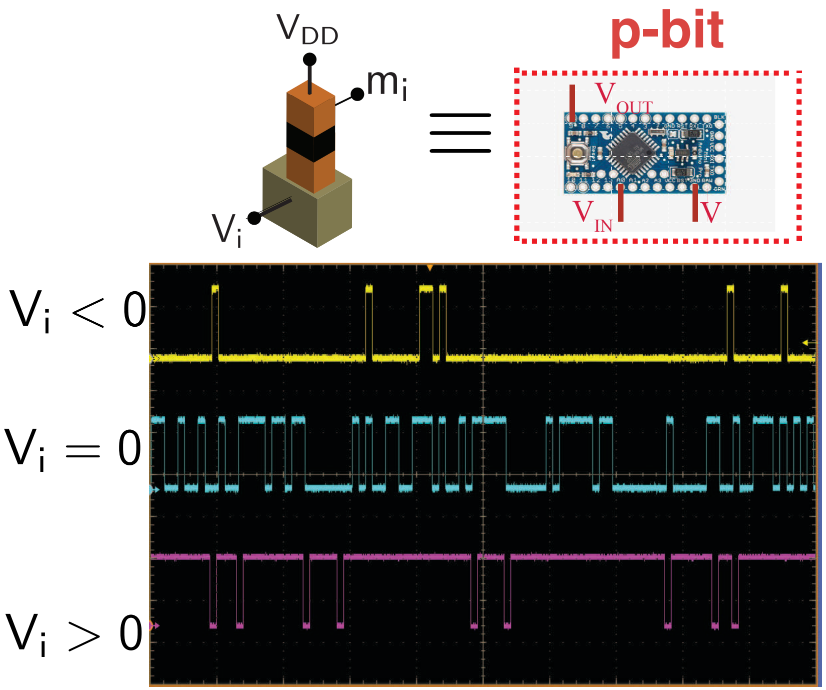




Borders et al., *Nature (2019)*

Non-magnetic p-computers

Arduino emulation

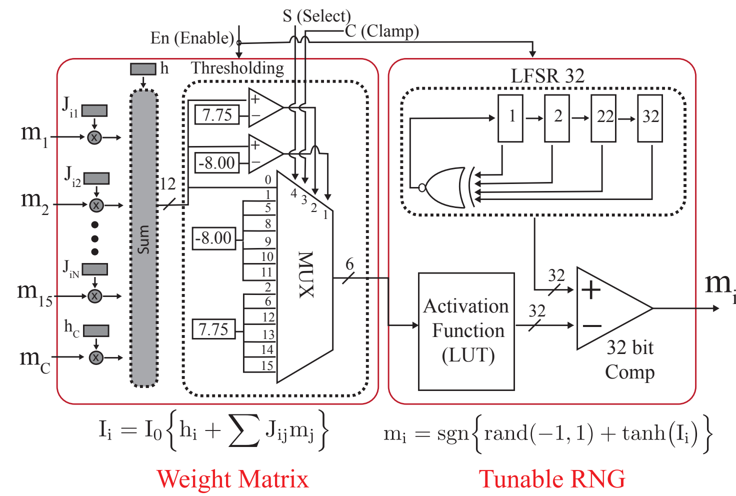


Pervaiz et al., *Sci. Rep. (2017)*

- **Autonomous (No global clocks)**

p-circuits up to ~50 p-bits

FPGA implementations



Pervaiz et al., *IEEE Transactions on Neural Networks and Learning Systems (2018)*

- **Synchronous (Clocked & Sequenced)**

p-circuits up to ~500 p-bits

- **Autonomous** (Emulated)

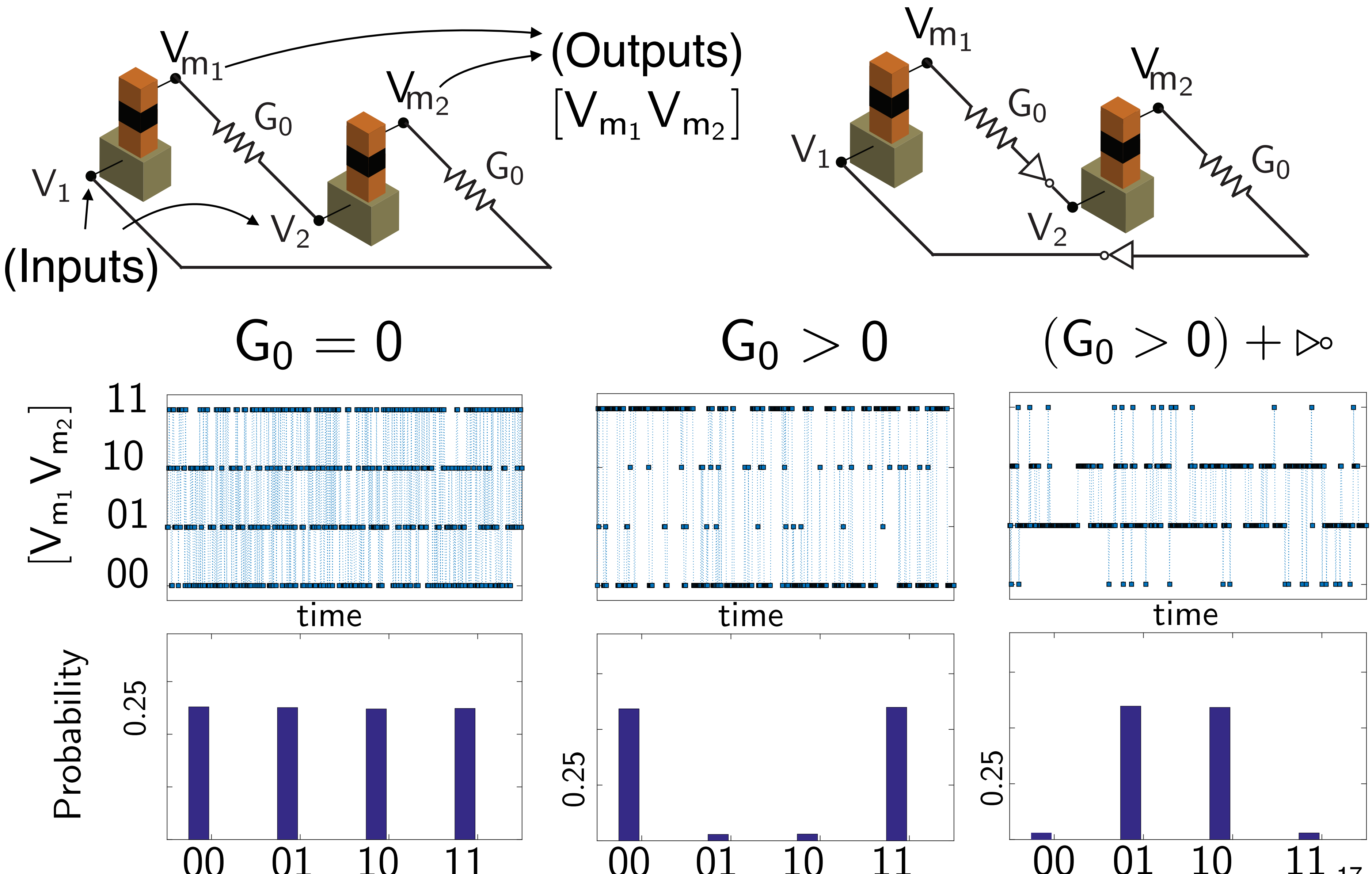
p-circuits up to **~8,000** p-bits

Sutton et al., *arXiv: 1907.09664 (2019)*

Outline

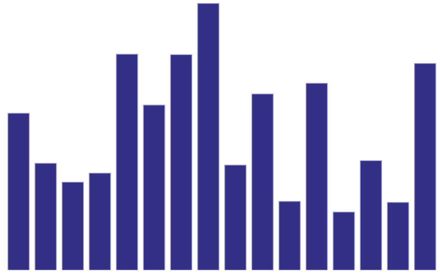
- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit
- Future directions

Correlated p-bits: p-circuits

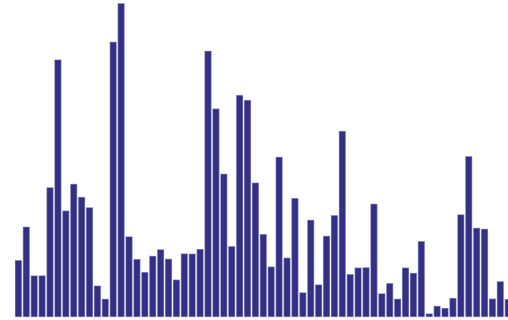


Correlated p-bits: p-circuits

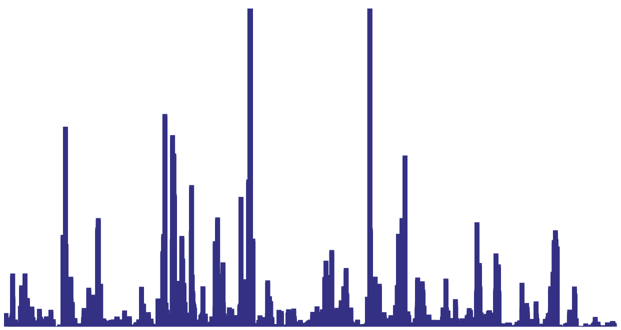
$N = 2$



$N = 4$



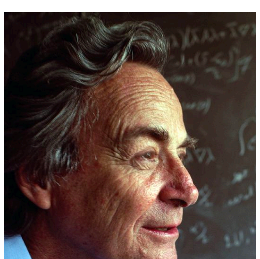
$N = 6$



$N = 8$

2^N states $\quad p_i \propto \exp(-E_i)$

$$E = -\sum_{i,j} J_{ij} m_i m_j - \sum_i h_i m_i$$



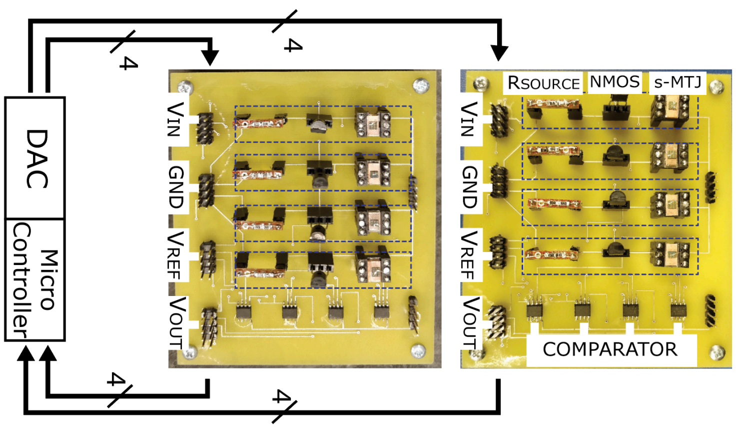
Feynman 1982 { *"the other way to simulate a probabilistic Nature,*
. . . is by a computer C, ***which itself is probabilistic****, . . .*
in which the output is not a unique function of the input"

Factorization as energy minimization

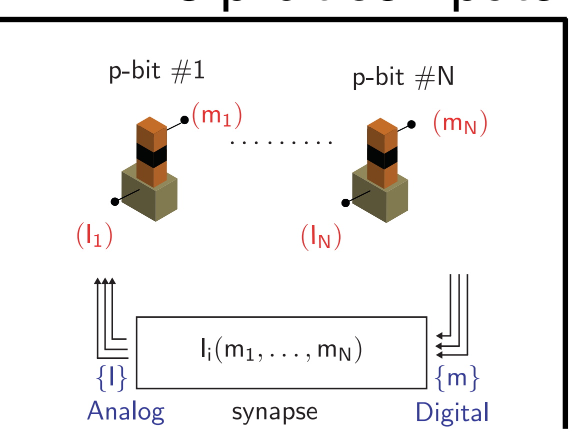
$$\text{minimize } E = (XY - F)^2$$

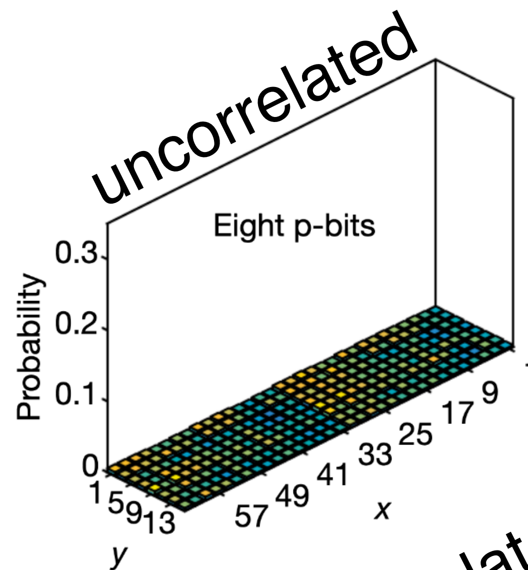
$$X = \sum_{p=0}^{P} 2^p x_p \; , \quad Y = \sum_{q=0}^{Q} 2^q y_q$$

$$I_i = -\frac{\partial E(x_1, .., x_N, y_1, .., y_N)}{\partial x_i}$$



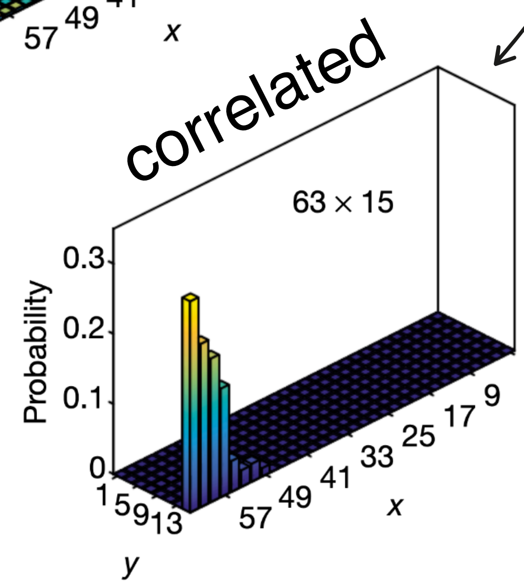
8 p-bit computer



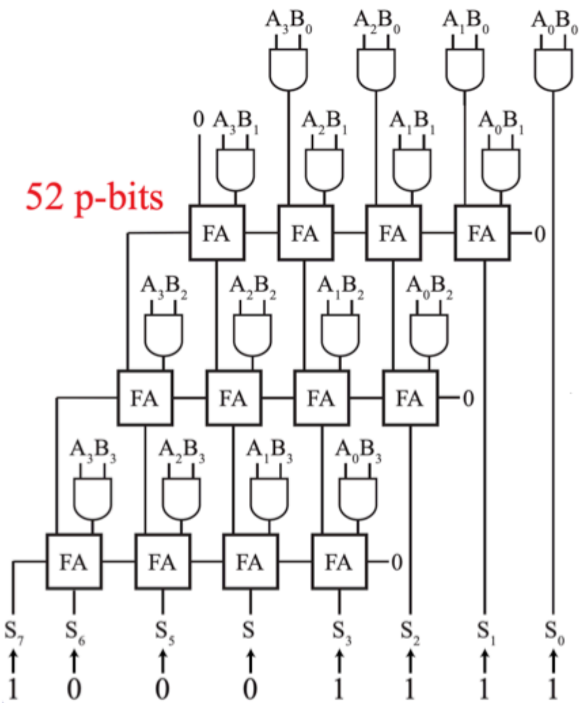


Borders et al., *Nature (2019)*

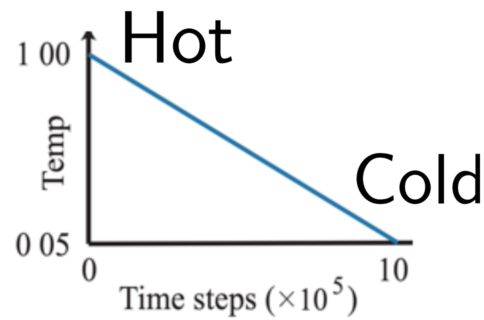
$$F = 945$$
$$= 63 \times 15$$



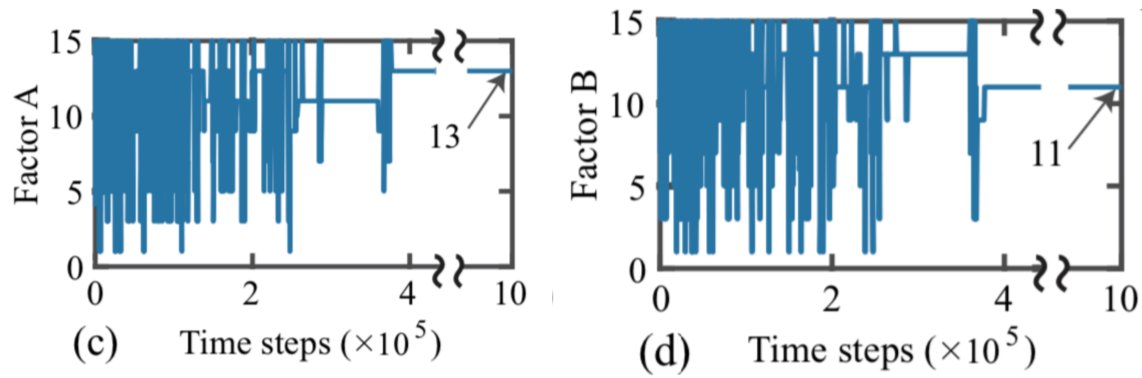
Factorization as Inverse Multiplication



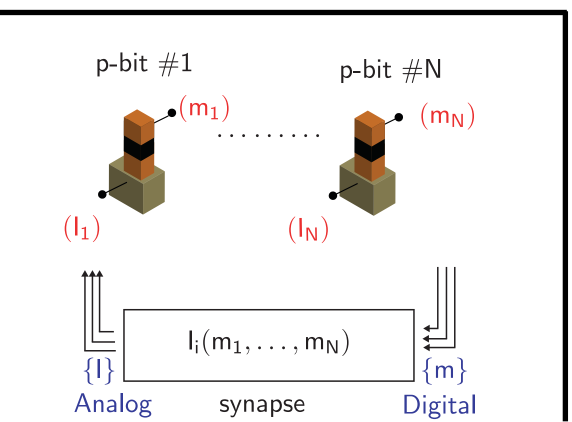
(143 in binary)



○ Electrical annealing
(Classical Annealing)

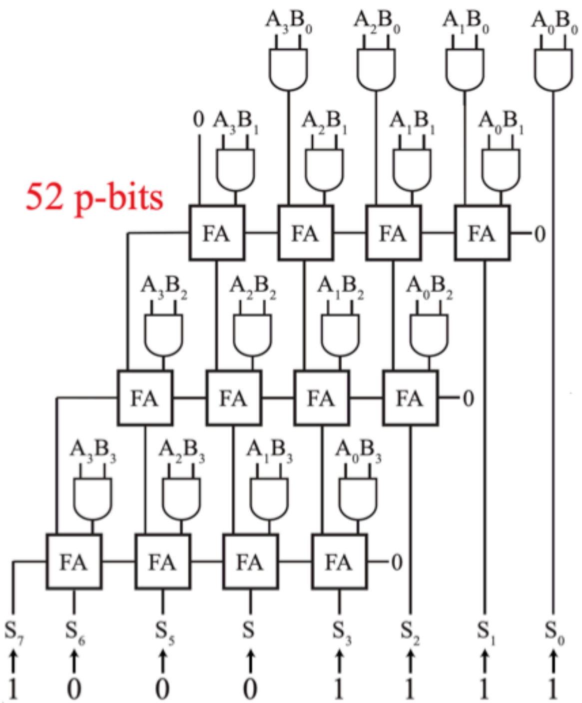


143 = 11 x 13



- Use VLSI synthesis to design cost functions for *any* inverse problem.
- Sparse graph, discrete and modular weights.

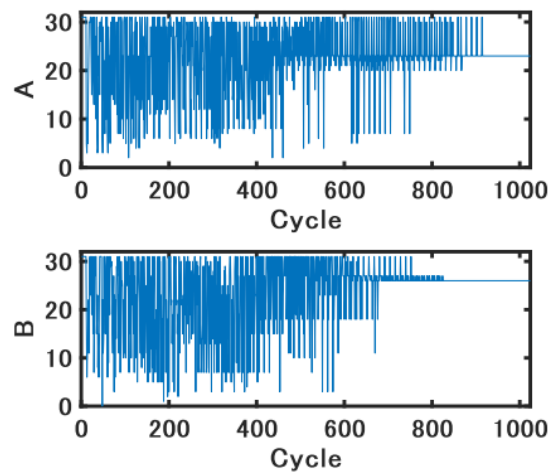
Factorization as Inverse Multiplication



Efficient CMOS Invertible Logic Using Stochastic Computing

Sean C. Smithson, *Student Member, IEEE*, Naoya Onizawa, *Member, IEEE*, Brett H. Meyer, *Senior Member, IEEE*, Warren J. Gross, *Senior Member, IEEE*, and Takahiro Hanyu, *Senior Member, IEEE*

IEEE TCAS-I (2019)



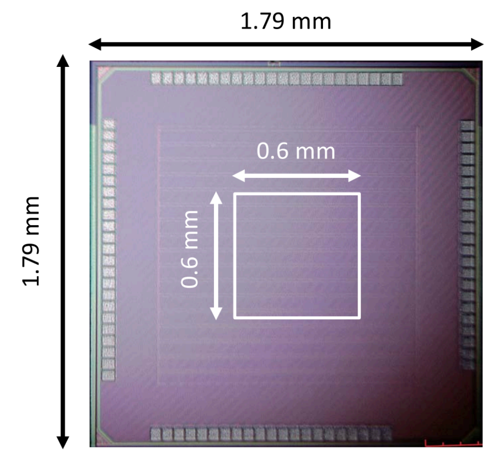
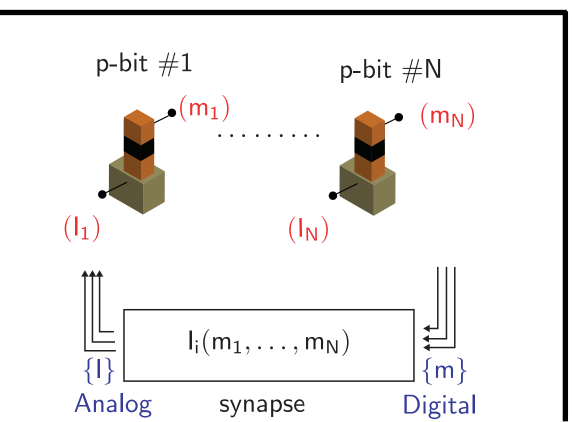


Fig. 14. Photomicrograph of fabricated 5-bit by 5-bit factorizer circuit.

65 nm CMOS chip



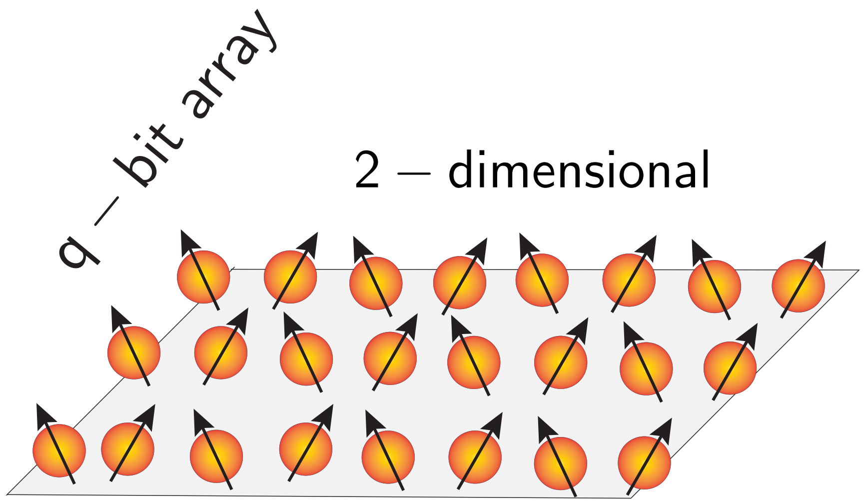
REFERENCES

[1] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, "Stochastic *p*-bits for invertible logic," *Phys. Rev. X*, vol. 7, no. 3, p. 031014, Jul./Sep. 2017.

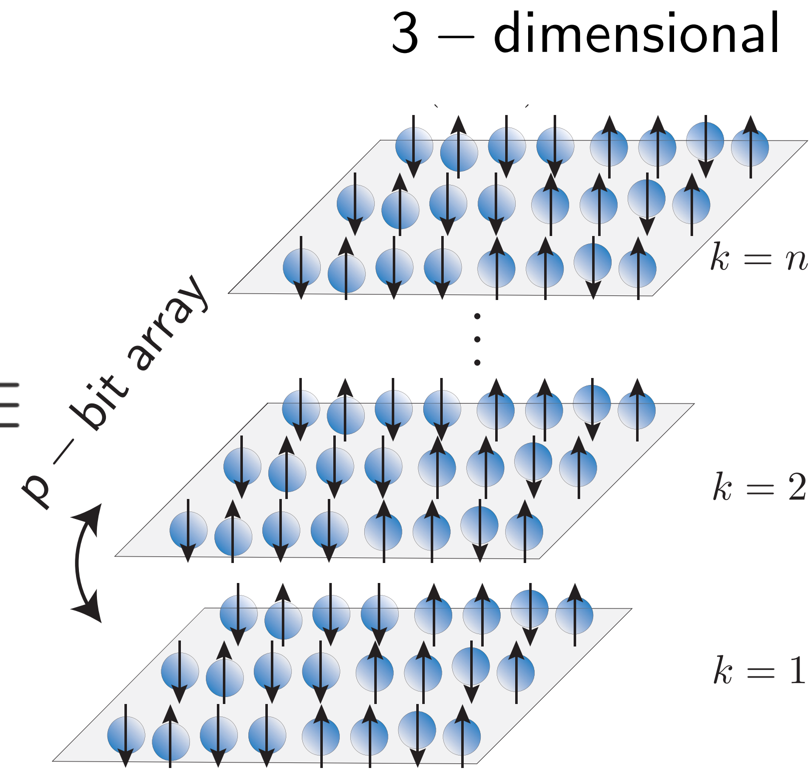
Outline

- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit
- Future directions

Adiabatic Quantum Computing (AQC)



≡



$$H_{\text{quantum}} = -\left(\sum_{i<j} J_{i,j}\sigma_i^z\sigma_j^z + \Gamma_x \sum_i \sigma_i^x\right)$$

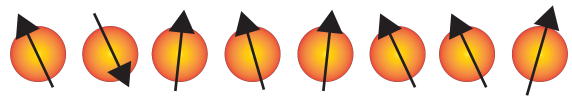
$$H_{\text{classical}} = -\left(\lim_{n\to\infty}\sum_{k=1}^{n}\sum_{i<j}(J_{\|})_{i,j}\ m_{i,k}m_{j,k} + J_{\perp}m_{i,k}m_{i,k+1}\right)$$

$$J_{\|} = J/n \quad J_{\perp} = -\frac{1}{2\beta}\log\tanh(\beta\Gamma/n)$$

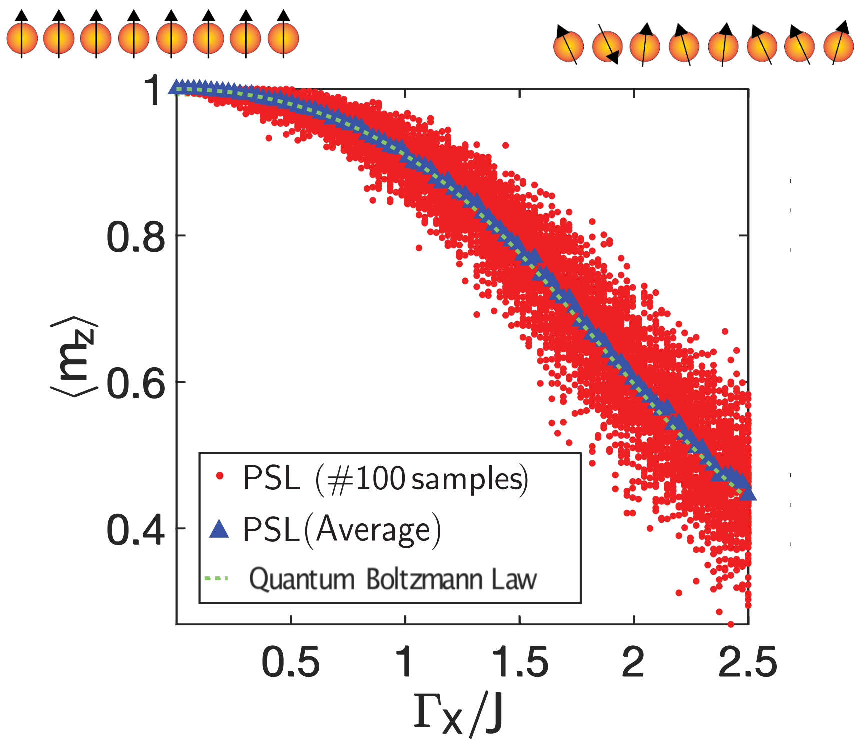
Camsari et al., *Physical Review Applied (2019)*

- Basis of well-known Quantum Monte Carlo methods in **software**
- Efficient for a subclass of quantum systems: Sign-problem free (*Major qualifier!*)

Mapping quantum systems to p-bits



N=8 q-bits (J=+1, **FM**)



$$H_{\text{quantum}} = -\left(\sum_{i<j} J_{i,j}\sigma_i^z\sigma_j^z + \Gamma_x \sum_i \sigma_i^x\right)$$

$$\langle m^z \rangle = \frac{\text{Tr.}[m^z \ \exp(-\beta H)]}{\text{Tr.}[\ \exp(-\beta H)]}$$

- Quantum Boltzmann Law

250 replicas

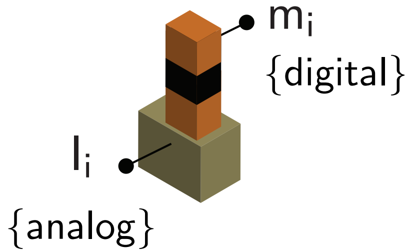
$$H_{\text{classical}} = -\left(\lim_{n\to\infty} \sum_{k=1}^{n} \sum_{i<j} (J_{\|})_{i,j} \ m_{i,k}m_{j,k} + J_{\perp} m_{i,k}m_{i,k+1}\right)$$

- Averages ***and*** correlations match

Outline

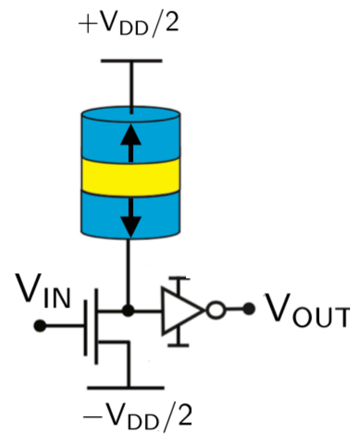
- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit
- Future directions

Device-level comparison

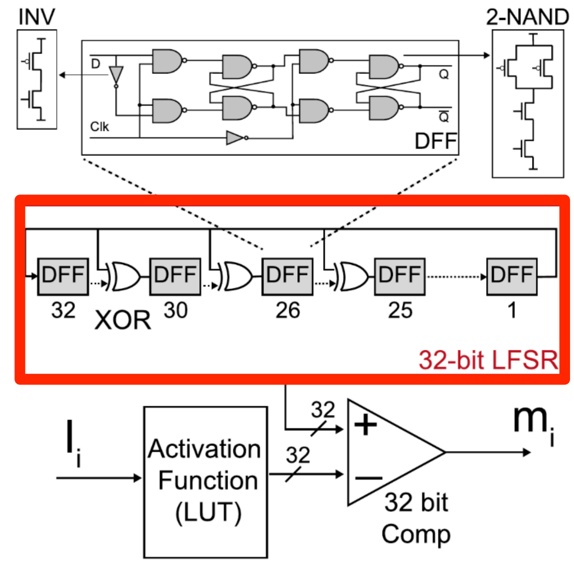


- MTJ design much smaller than CMOS LFSR
- Conservative energy analysis:
  1. Parasitics, clock power **not** included
  2. CMOS LFSR **not** tunable by itself
- MTJ gives True RNG LFSR gives Pseudo RNG

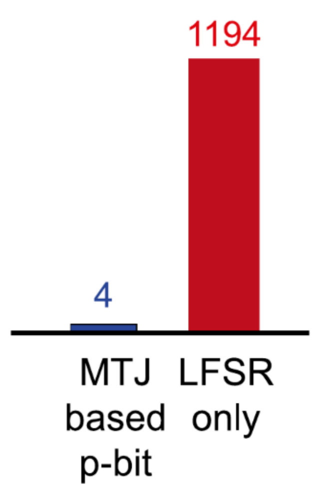
MTJ-based



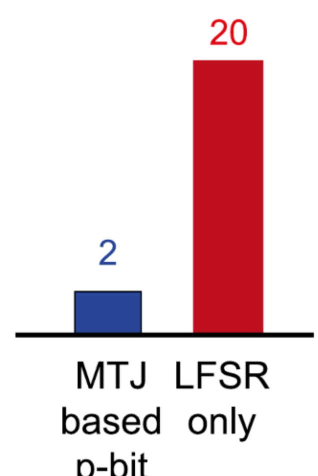
CMOS-based



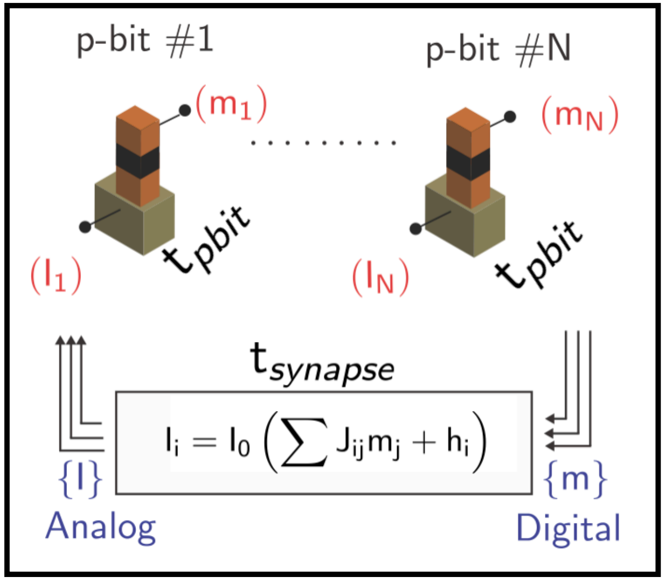
Transistor Count (#)



Energy per random bit (fJ)



Architecture-level comparison



- Key Metric:
  Number of (informed) flips per second

$$\tau_{\text{synapse}} < \tau_{\text{pbit}}$$

- How many flips?

Problem / Algorithm dependent

$$\tau = \tau_0 \, f(N)$$

- Best performance with optimized state-of-the-art GPU / CPU

  $f \leq 10^{12}$ flips/second

$N = 10^6$ p-bits

$t_{pbit} = 100$ ps (Projected)

$f \geq 10^{16}$ flips/second

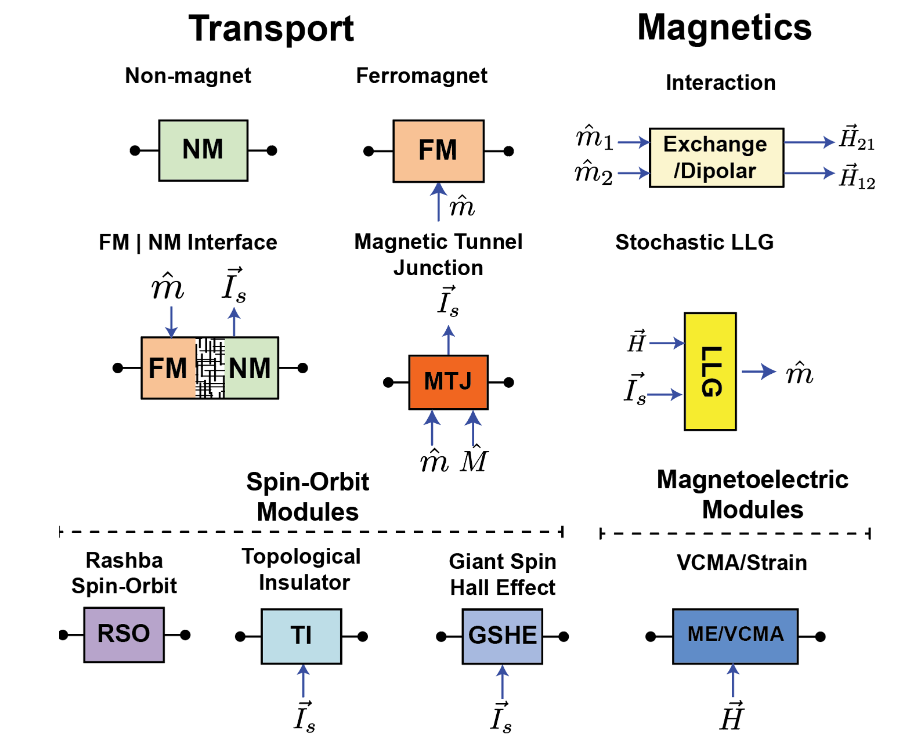
- Why not 10^9 p-bits?
  - Power wall?
  - Area wall?
  - Interconnect wall?

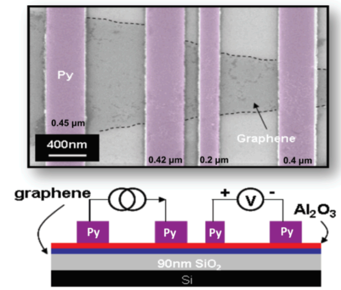
1Gb MRAM

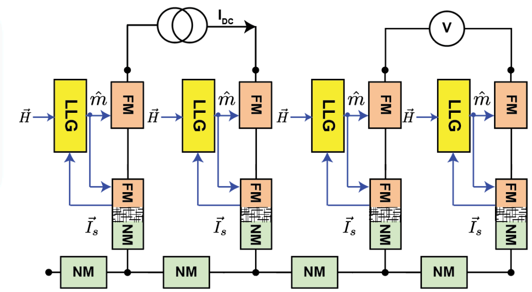
Outline

- Introduction: p-bits and p-circuits
- How to build a p-bit
- Example p-circuits
- p-bit vs. q-bit
- p-bit vs. bit
- Future directions

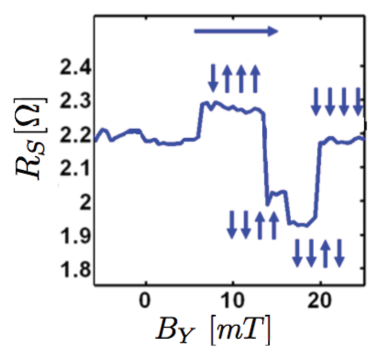
Future Directions: Materials / Devices



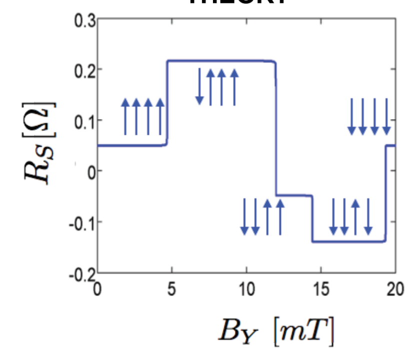




EXPERIMENT



THEORY



Spin-Transfer-Torque

STT-Oscillators

Valley-Hall Effect

Antiferromagnets (Synthetic / Natural)

RRAM / Memristors

Skyrmions

Domain wall motion

Ferroelectrics

Cauldron of emerging phenomena

Spin-Hall Materials

Spin-Orbit-Torque

Voltage-control of magnetism (VCM)

TMDs

Graphene

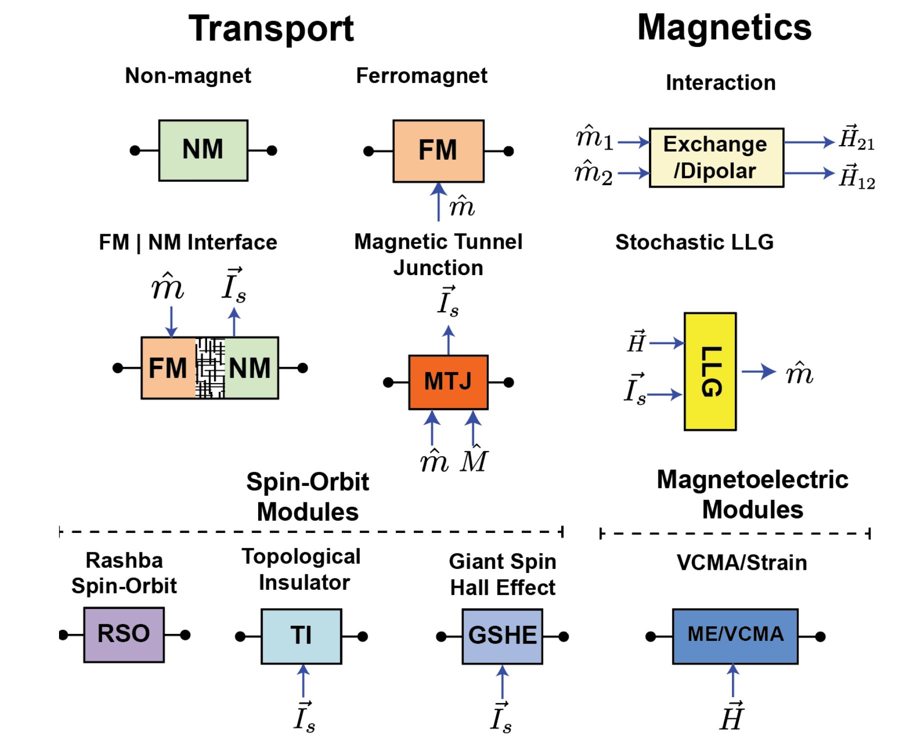
Phase-change materials

Topological Insulators

2D Magnetism

Multiferroics

Future Directions: Materials / Devices



- Enabled transition from Materials/Devices to Circuits/Systems in the context of p-bits
- CS expert with no knowledge of magnetics is able to run 100+ magnet p-bit networks in a day
- A colleague recently: "We use it *every day*"

Spin-Transfer-Torque

RRAM / Memristors

STT-Oscillators

Skyrmions

Valley-Hall Effect

Domain wall motion

Antiferromagnets (Synthetic / Natural)

Ferroelectrics

Cauldron of emerging phenomena

Spin-Hall Materials

Spin-Orbit-Torque

Voltage-control of magnetism (VCM)

TMDs

Graphene

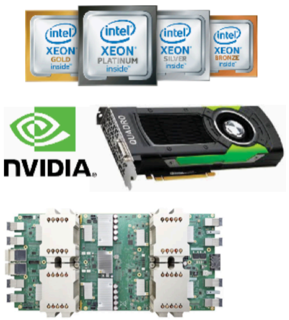
Phase-change materials

Topological Insulators

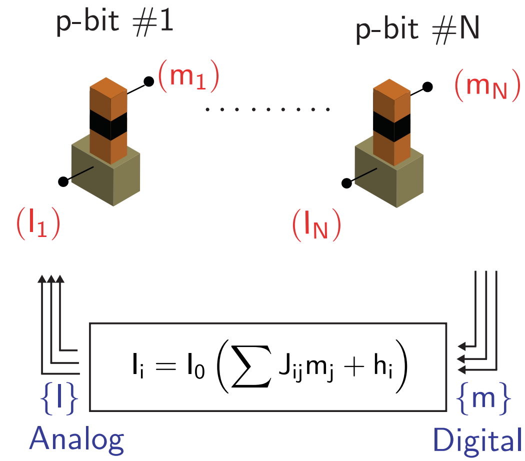
2D Magnetism

Multiferroics

Future Directions: Circuits / Systems / Algorithms



Machine Learning



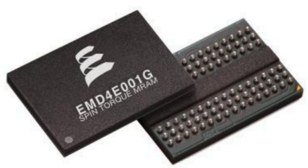
NISQ-era QC

- Implementing p-bits: FPGA / CMOS ASIC / MRAM-based / Emerging devices?

- Define p-bit supremacy?

$$f = 10^{16} \text{ flips/s}$$

$$f_{CPU} = 10^{12} \text{ flips/s}$$



1Gb MRAM

(manufacture-ready)

- Which problems or algorithms to accelerate?

Look for **naturally stochastic** ones!

Machine Learning and Optimization are full of examples

bits, p-bits, q-bits

bits

$\tau_0 = 10^{12}$ flips/s

p-bits

$\tau_0 = 10^{16}$ flips/s

$$\tau = \tau_0 \; f(N)$$

q-bits

f(N) :

$\exp(N) \rightarrow \log(N)$

Shor's algorithm

Grover's Search

Fault-tolerant, scaled QC

Closing the loop

Probabilistic Simulation of Quantum Circuits: Applications & Limitations
in preparation

bits, p-bits, q-bits

bits

p-bits

q-bits

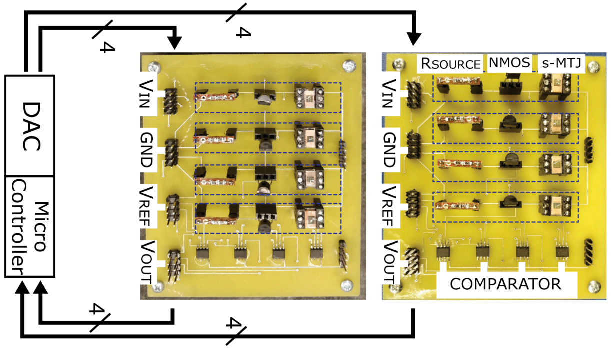
NISQ-era QC

- No established "Shor-like" improvements: $\exp(N) \rightarrow \log(N)$

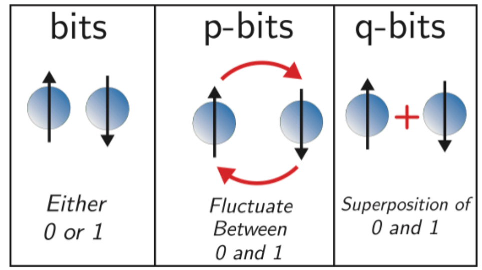
- Quantum Machine Learning
- Quantum internet
- Optimization (QAOA)

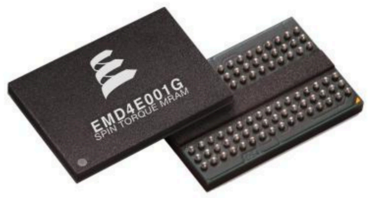
NISQ-era: *Some* applications resemble "flying cars" rather than airplanes

Summary



8 p-bit computer ✓



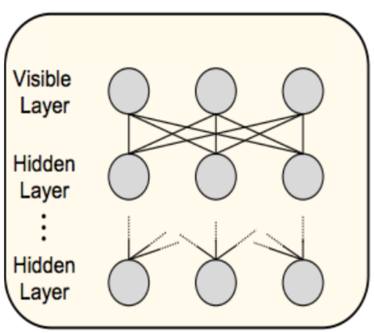


1 Mb MRAM-based p-computer ?

FPGA — Digital CMOS — eMRAM — Emerging Devices ?

Machine Learning & Quantum Computing

Learning & Inference



Invertible Logic

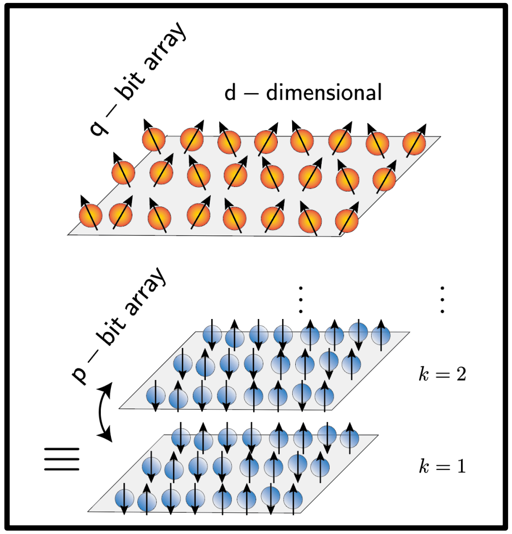
Adder ⟷ Subtractor

(1) Subset Sum Problem

Multiplier ⟷ Divider

(2) Prime Factorization

Quantum Annealing

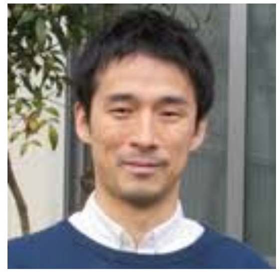


Collaborators

Supriyo Datta*

Hideo Ohno (Tohoku U)

Shunsuke Fukami (Tohoku U)

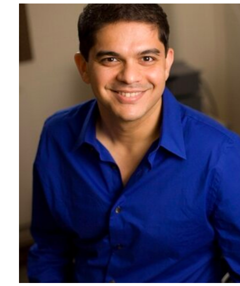


Zhihong Chen*

Joerg Appenzeller*

Sayeef Salahuddin (UC Berkeley)

Sunil Bhave*



Drew Borders (Tohoku U)

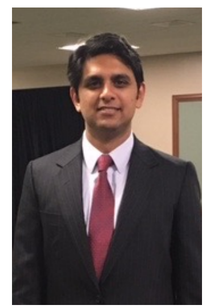
Rafatul Faria (Intel)

Orchi Hassan*

Brian Sutton (Microsoft)

Punyashloka Debashis (Intel)

Zeeshan Pervaiz (Intel)



Jan Kaiser*

Shuvro Chowdhury*

Anirudh Ghantasala*

Behtash Behin-Aein (Globalfoundries)

Giovanni Finocchio (UofMessina)

Mert Torunbalci (Microsoft)

Pramey Upadhyaya*

Samiran Ganguly (UVA)

Ramtin Zand (UofSC)

Tingting Shen*

Vaibhav Ostwal*

*Affiliations: Purdue University