p-bits for Probabilistic Spin Logic (PSL)

Supriyo Datta

PURDUE UNIVERSITY

Jan Kaiser

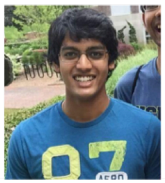
Zeeshan Pervaiz

Brian Sutton

Dr. Kerem Camsari

Shuvro Chowdhury

Aniruddh Ghantsala

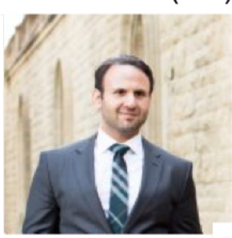


Orchi Hassan

Rafatul Faria

Collaborators

Behtash Behin-Aein (GF)



Sayeef Salahuddin (UC Berkeley)

Punya Debashis

Joerg Appenzeller

Zhihong Chen

Ernesto Marinero

[https://arxiv.org/abs/](https://arxiv.org/abs/1809.04028) 1809.04028

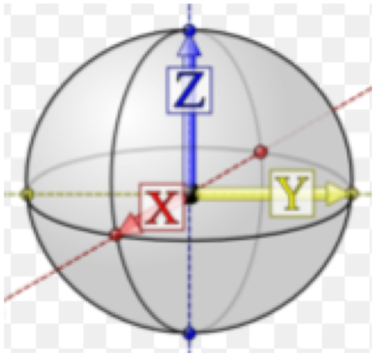
Sponsors

SRC

DARPA

NSF

1a. Bits & q-Bits



Single spins

delicate superposition of 0 & 1

either 0 or 1

Bits

q-bits

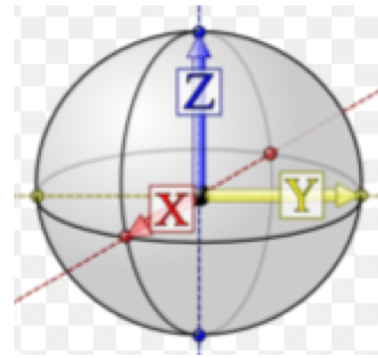
Digital computing

Quantum computing

1b. Bits & q-Bits

Hard disks
MRAM / MTJ's
Stable magnets

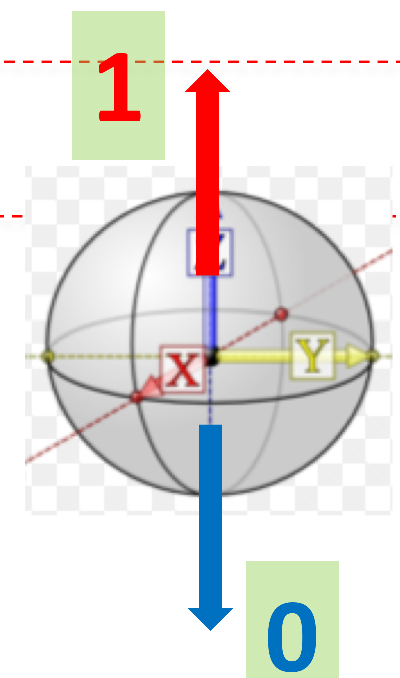
either
0 or 1



Single spins

delicate
superposition
of 0 & 1

Bits



q-bits

Digital
computing

Quantum
computing

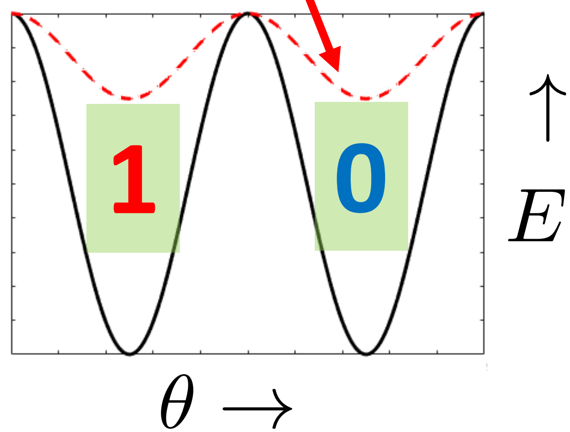
1c. p-Bits

Unstable magnets

Hard disks
MRAM / MTJ's

Stable magnets

Single spins



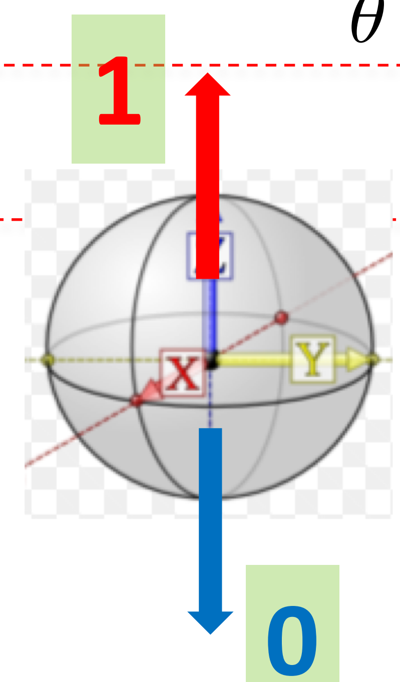
either
0 or 1

delicate superposition of 0 & 1

Bits

q-bits

Digital computing



Quantum computing

1d. p-Bits

Hard disks
MRAM / MTJ's

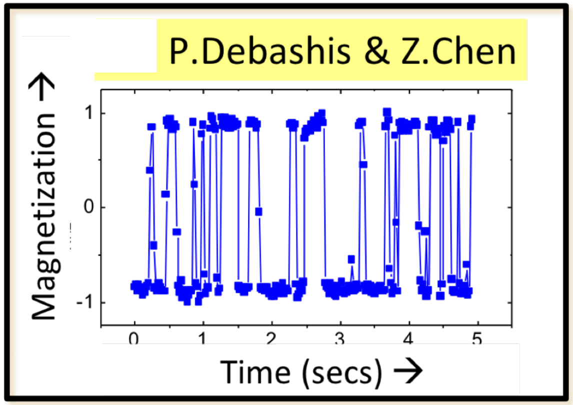
Stable magnets

either
0 or 1

Bits

Digital
computing

Unstable
magnets



p-bits

p- circuits
p- computing

Single
spins

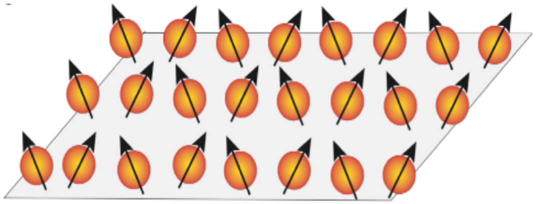
delicate
superposition
of 0 & 1

q-bits

Quantum
computing

2a. Adiabatic Quantum Computing

q-bits



$$H = \sum_{i,j} J_{ij} \; \vec{\sigma_i} \cdot \vec{\sigma_j}$$

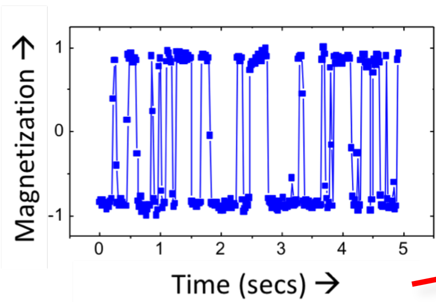
Optimization
Problems

E → Cost Function

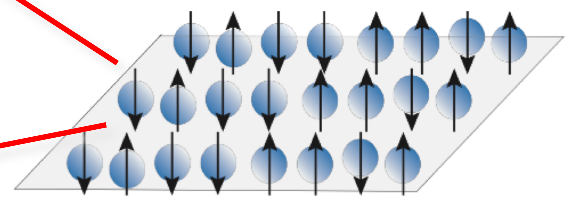
2b. Probabilistic Spin Logic (PSL)

Adiabatic Quantum Computing

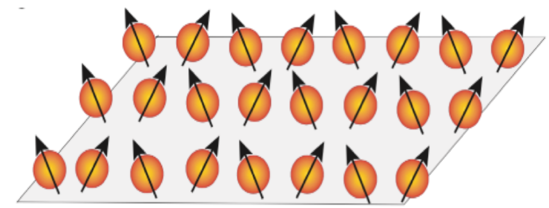
Unstable magnet



p-bits



q-bits



$$H = \sum_{i,j} J_{ij}\ \vec{\sigma_i} \cdot \vec{\sigma_j}$$

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

Optimization Problems

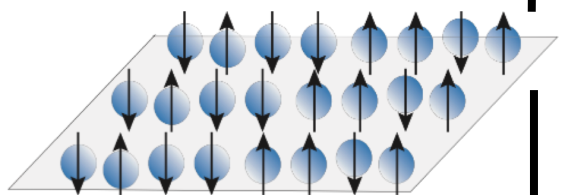
E → Cost Function

2c. Correlating versus Entangling

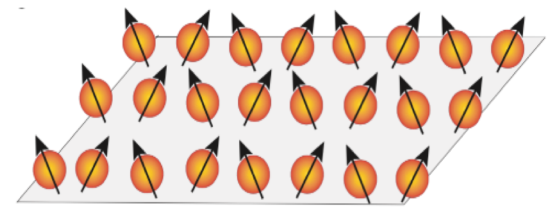
Probabilistic Spin Logic (PSL)

Adiabatic Quantum Computing

p-bits



q-bits



$$E = \sum_{i,j} J_{ij} \; m_i \; m_j$$

$$H = \sum_{i,j} J_{ij} \; \vec{\sigma_i} \cdot \vec{\sigma_j}$$

Feynman 1982

Boltzmann Law

$$P(m_1, \cdot\cdot, m_N) \sim e^{-E}$$

- Classical many-body system

Classical Interaction

- Quantum many-body system

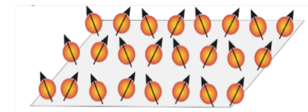
Coherent Interaction

3a. SNN – PSL -- AQC

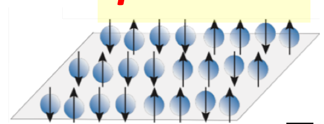
Adiabatic Quantum Computing

Stochastic Neural Networks

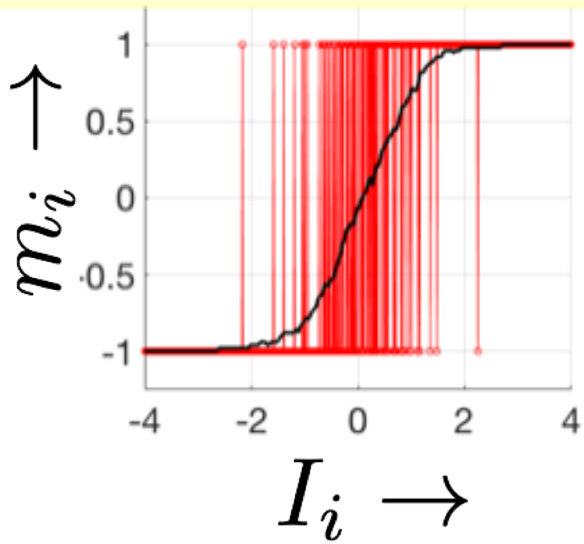
q-bits



p-bits



Binary Stochastic Neuron (BSN)



$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

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

3b. SNN – PSL

Stochastic
Neural Networks

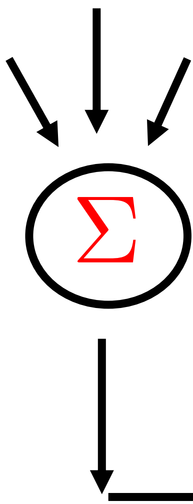
$$m_i = \mathrm{sgn}\left[\tanh I_i - rand(-1,+1)\right]$$

Binary Stochastic
Neuron (BSN)

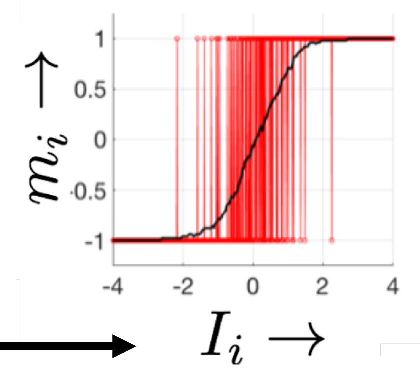
p-bits

"synapse"

m_j



$I_i(m_1 \cdot\cdot\, m_N)$



$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

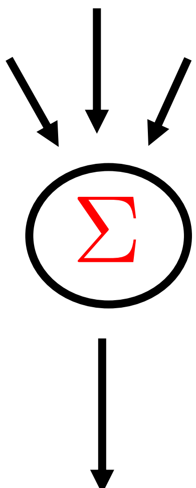
3c. SNN – PSL

Stochastic Neural Networks

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

"synapse"

m_j



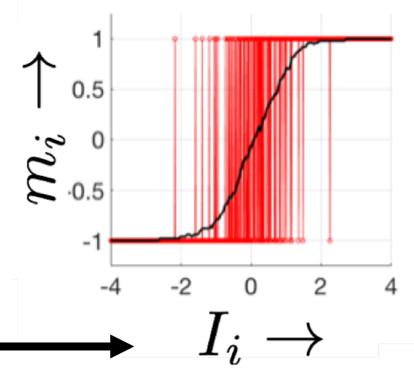
For a given E

$$I_i(m_1 \cdot\cdot m_N)$$

$$= -\frac{\partial E(m_1, \cdot\cdot, m_N)}{\partial m_i}$$

Binary Stochastic Neuron (BSN)

p-bits



$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

3d. Boltzmann Machines

→ Machine Learning

"synapse"

m_j

Σ

Boltzmann Law

$$P(m_1, \cdot\cdot, m_N) \sim e^{-E}$$

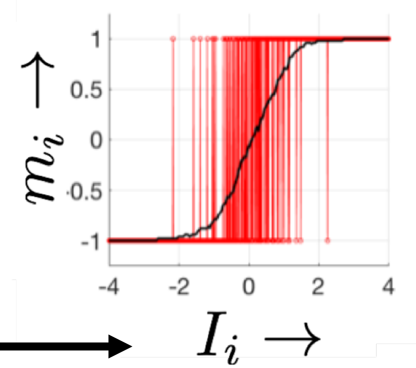
For a given E

$$I_i(m_1 \cdot\cdot m_N)$$

$$= -\frac{\partial E(m_1, \cdot\cdot, m_N)}{\partial m_i}$$

Binary Stochastic Neuron (BSN)

p-bits



$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

3e. Boltzmann Machines

→ Machine Learning

"synapse"

m_j

Σ

Boltzmann Law

$$P(m_1, \cdot\cdot, m_N) \sim e^{-E}$$

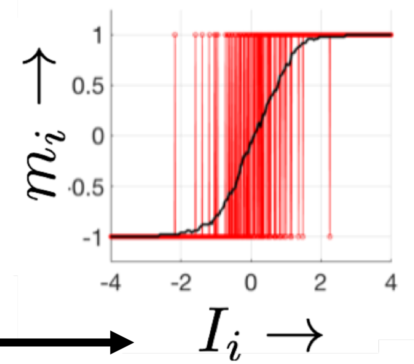
For a given E

$$I_i(m_1 \cdot\cdot m_N)$$

$$= -\frac{\partial E(m_1, \cdot\cdot, m_N)}{\partial m_i}$$

Binary Stochastic Neuron (BSN)

p-bits

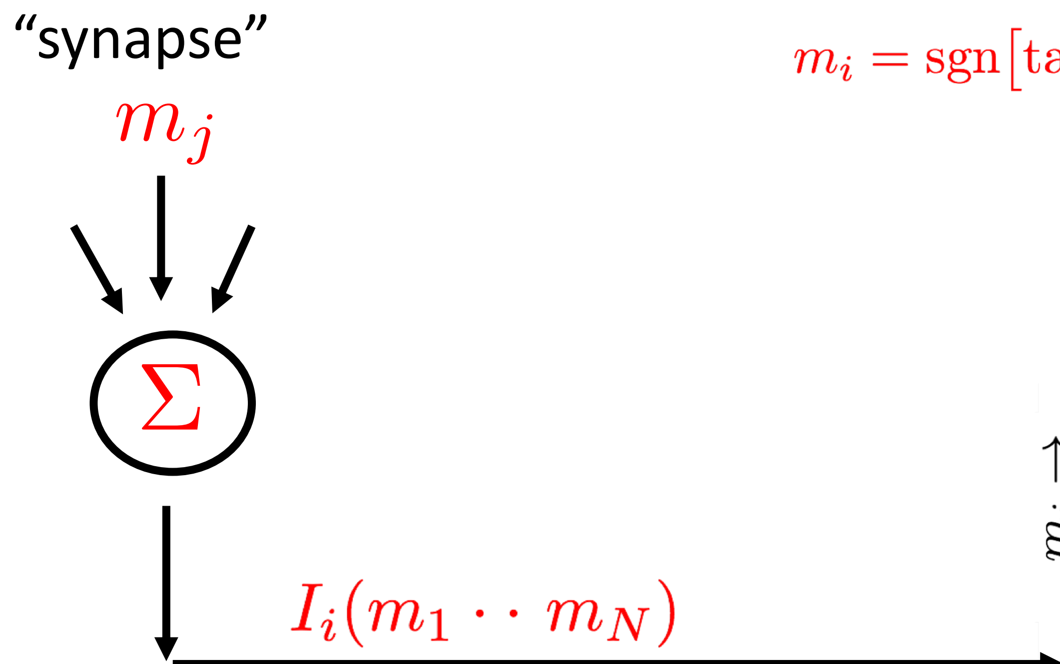


$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

Reciprocal Couplings $\boxed{\frac{\partial I_i}{\partial m_j} = \frac{\partial I_j}{\partial m_i} = -\frac{\partial^2 E}{\partial m_i \partial m_j}}$

4a. Why Hardware?

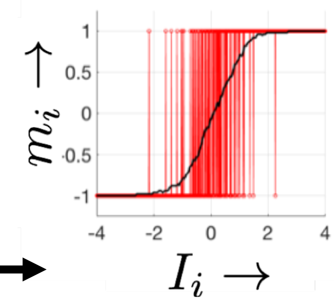
- Speed
- Power
- Area



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

Binary Stochastic Neuron (BSN)

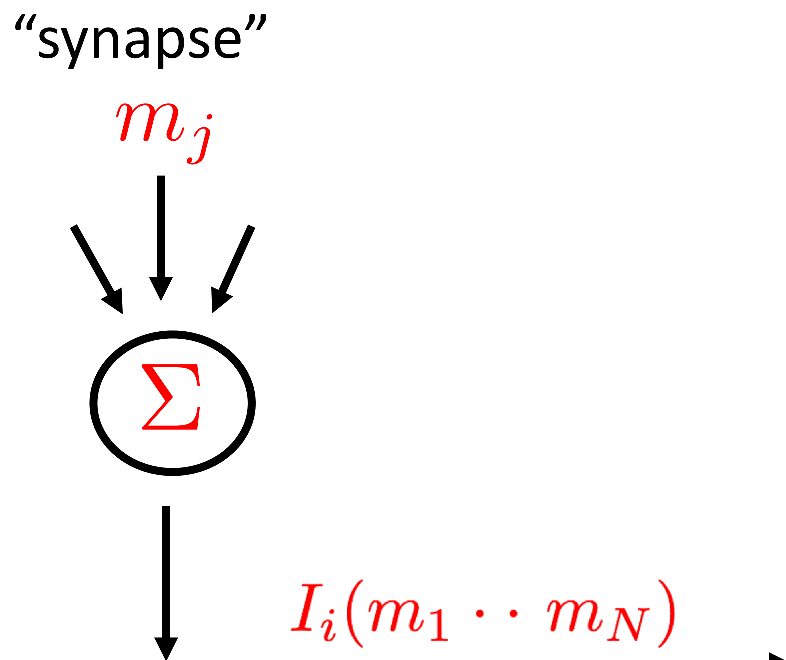
p-bits



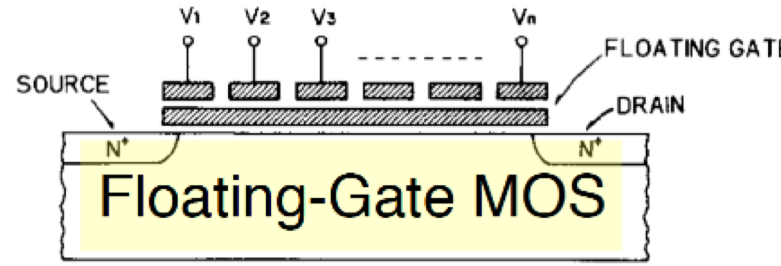
$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

4b. Hardware Synapse

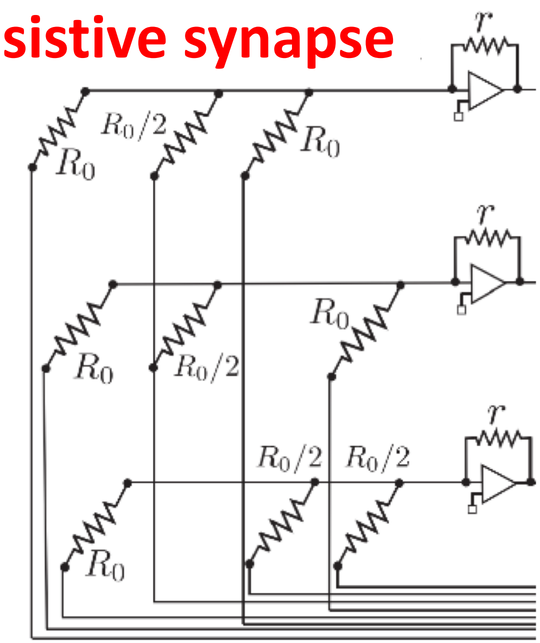
- **GPU**
- **FPGA**



- **Capacitive synapse**

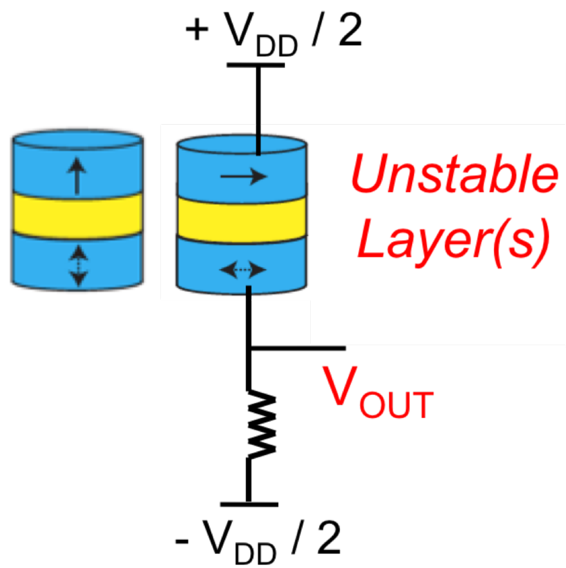


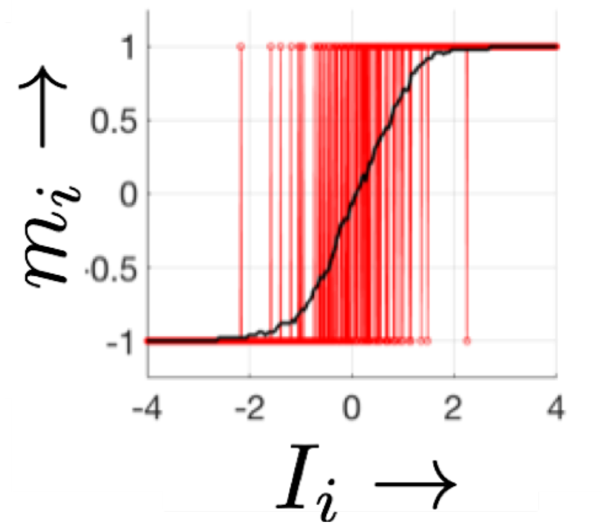
- **Resistive synapse**



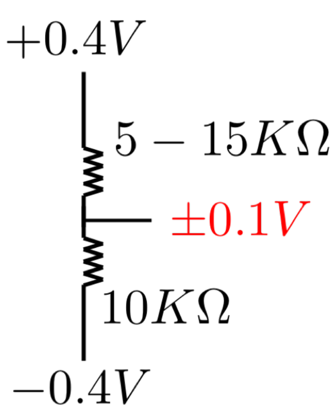
4c. Hardware Neuron

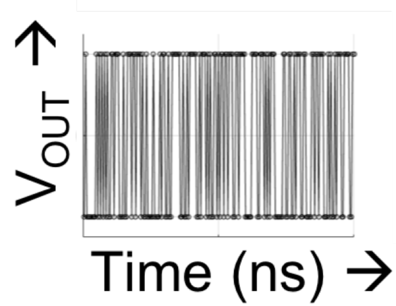
Magnetic Tunnel Junction (MTJ)





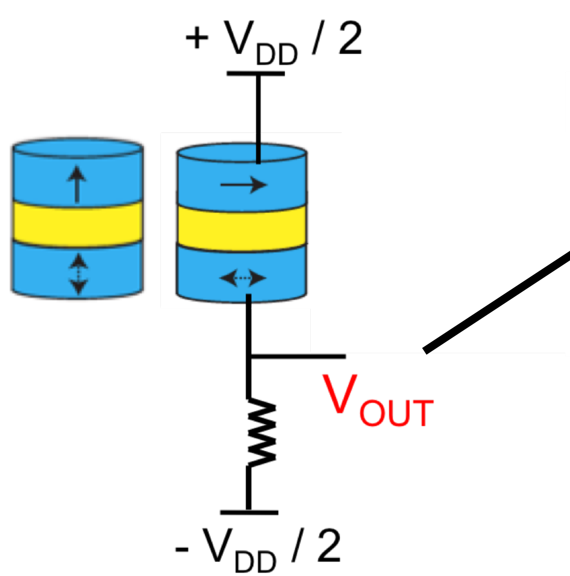
$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

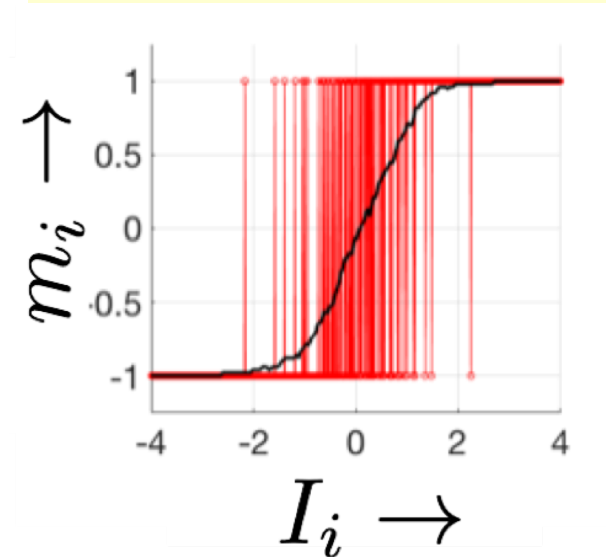




5a. Need third terminal

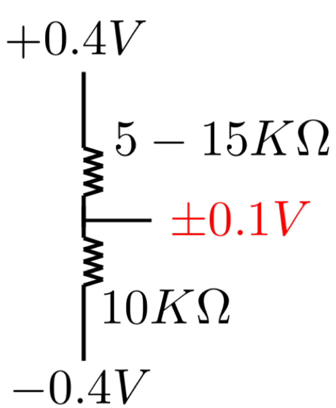
Binary Stochastic Neuron (BSN)



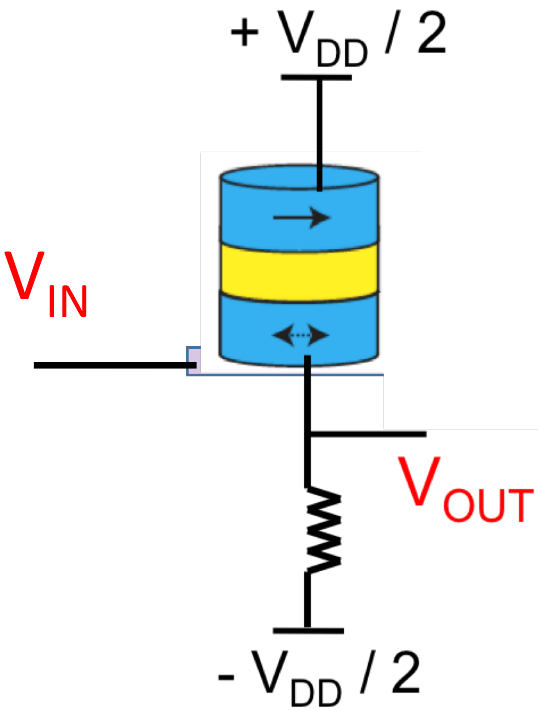


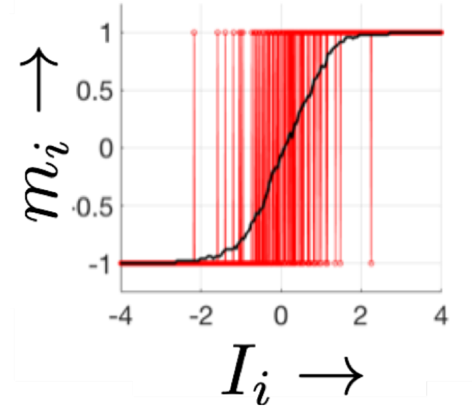
$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

V_{IN}



5b. Need third terminal





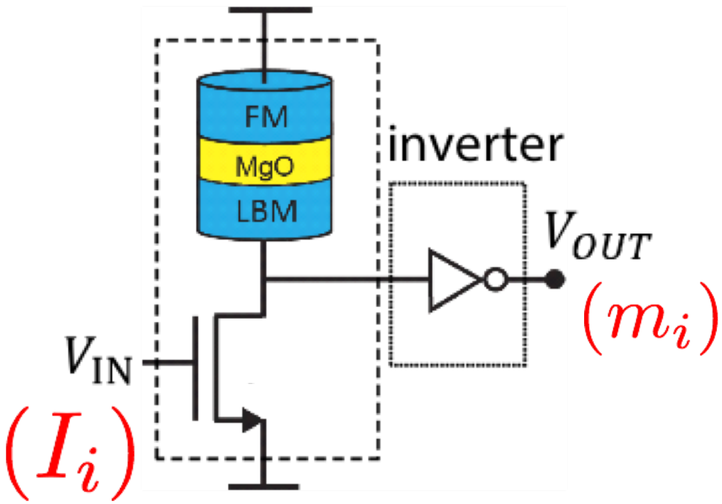
$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

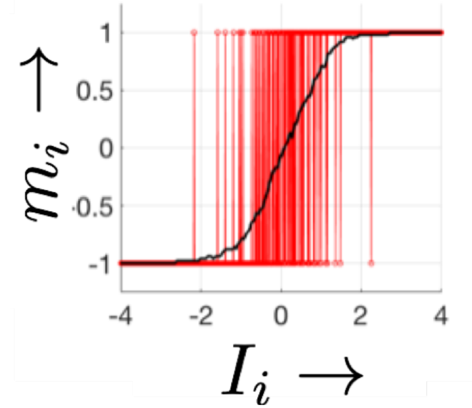
V_{IN}

Use V_{IN} to control magnet

- Spin-orbit Torque (SOT)
- Magnetoelectric Effect (ME)

5c. Need third terminal

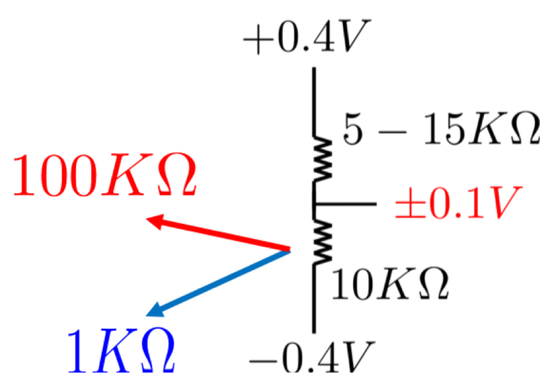




$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

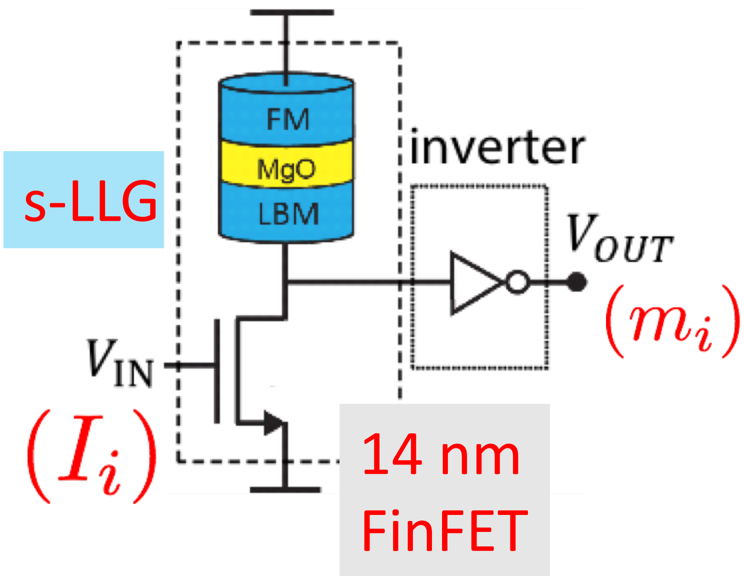
V_{IN}

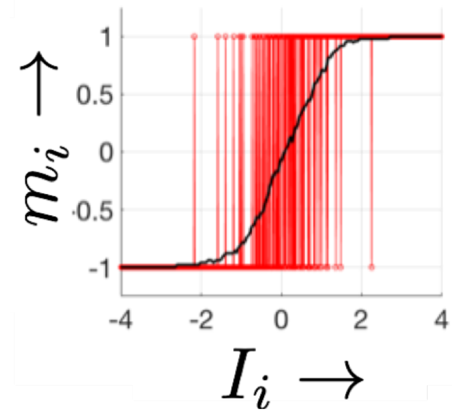
Use V_{IN} to control series resistance



→ amplify

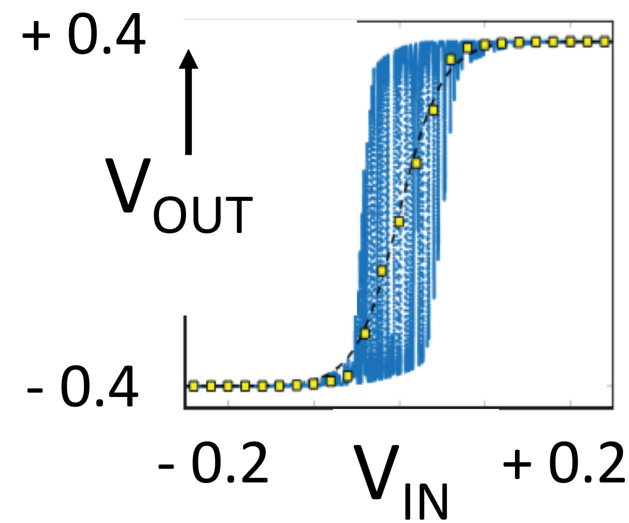
5d. Need third terminal





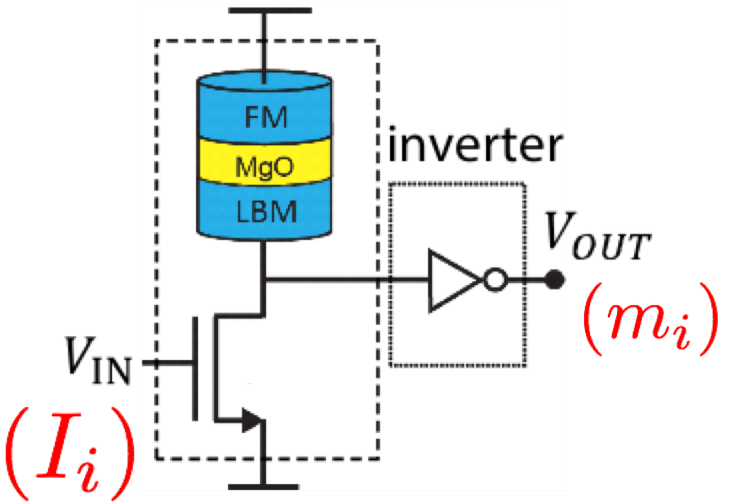
$$\frac{P(+1)}{P(-1)} = e^{2I_i}$$

Full SPICE simulation



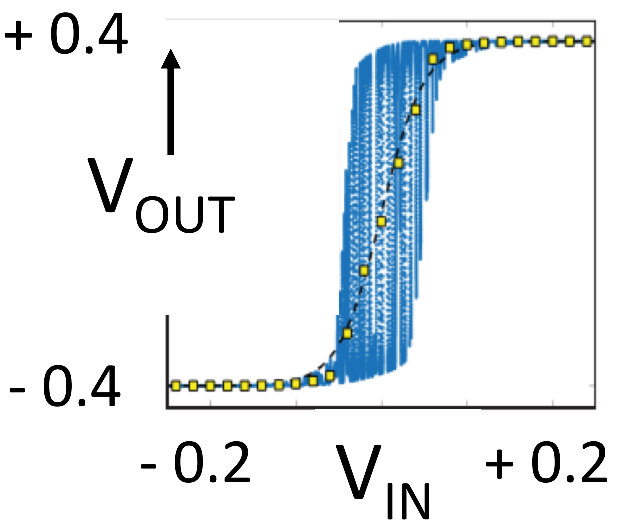
6a. 1T/MTJ: Embedded MRAM

Low Barrier Magnet (LBM)

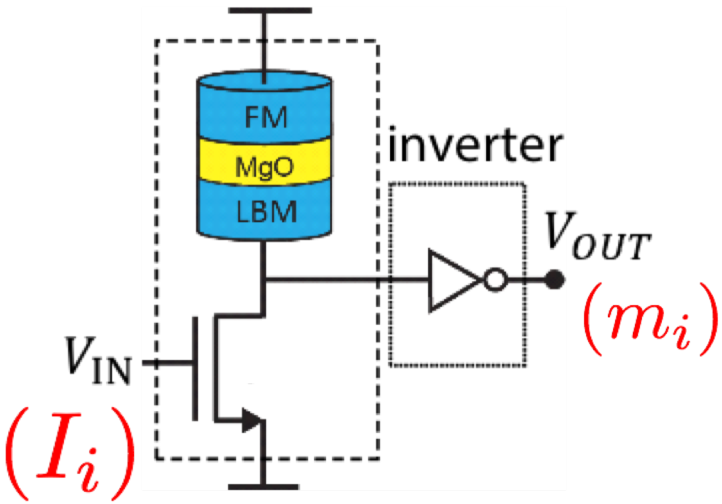


$\Delta \lesssim 15kT$

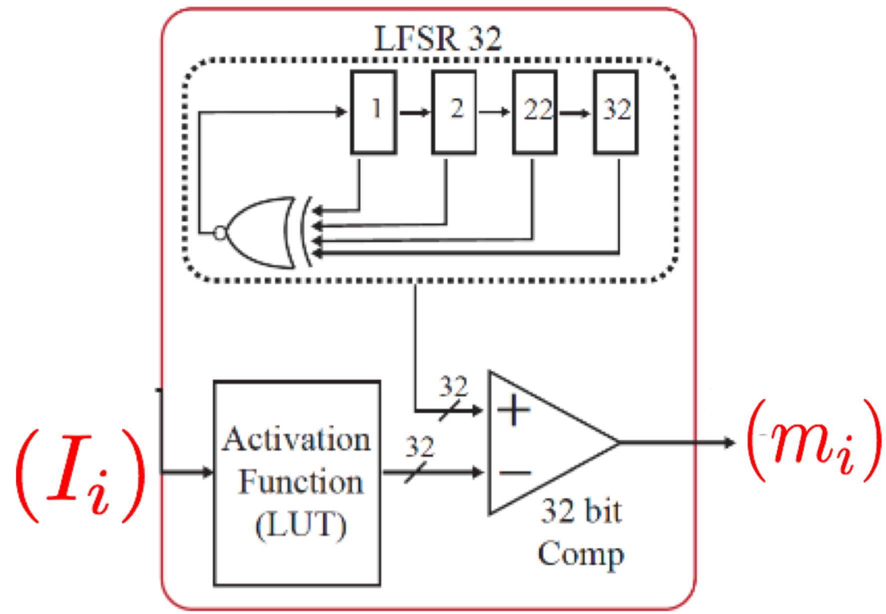
$$m_i = \mathrm{sgn}\left[\tanh I_i - rand(-1,+1)\right]$$



6b. 1T/MTJ: Embedded MRAM



Digital BSN

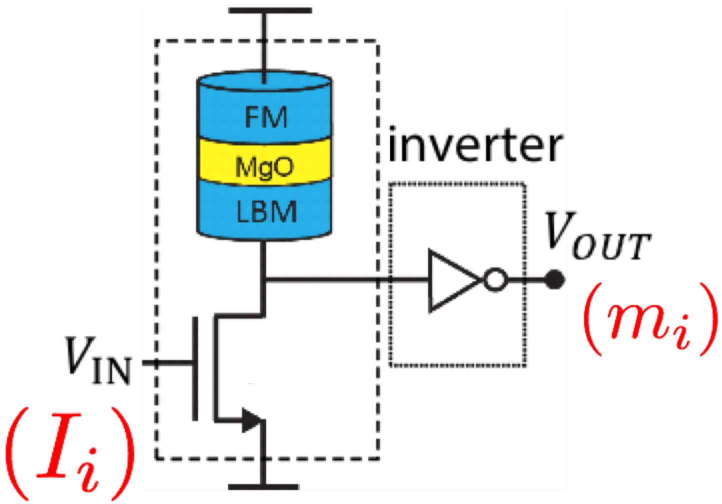


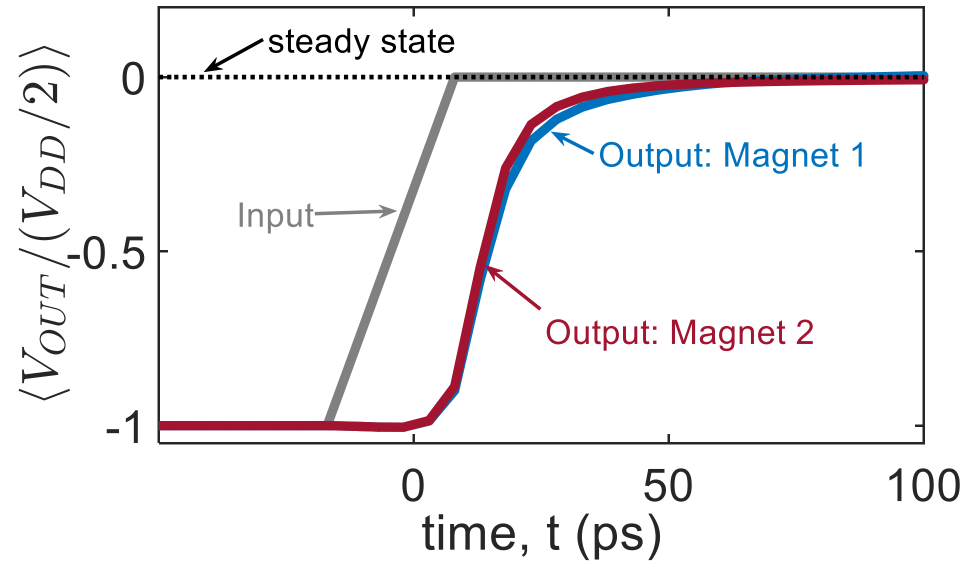
Hardware Accelerator

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

- Speed
- Power
- Area

6c. Speed

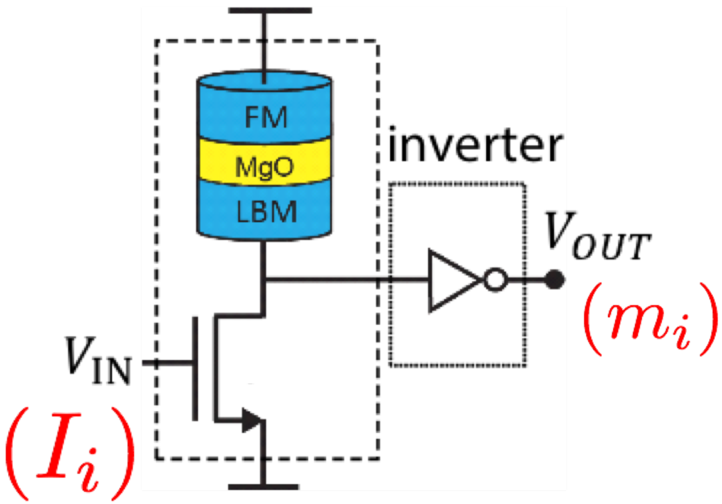


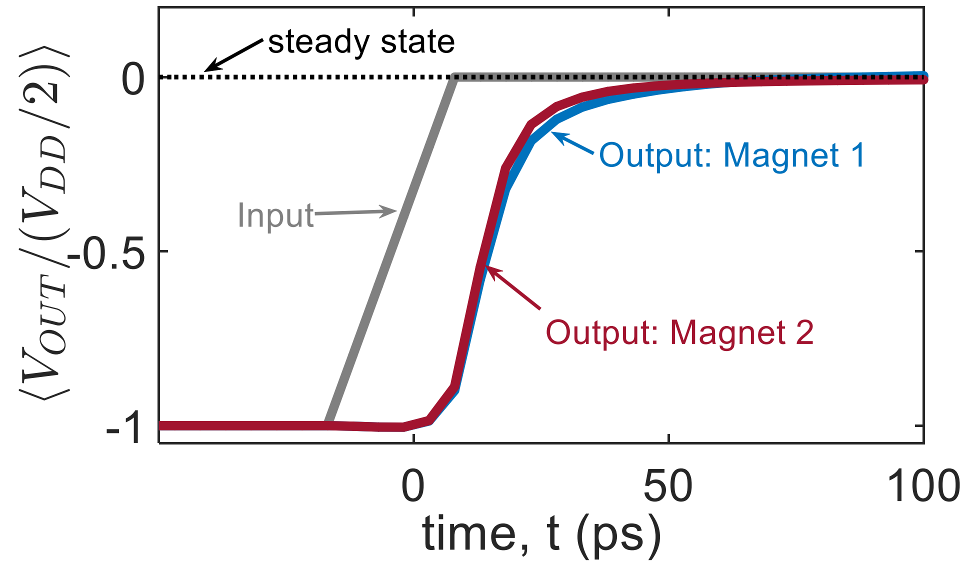


Hardware Accelerator

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

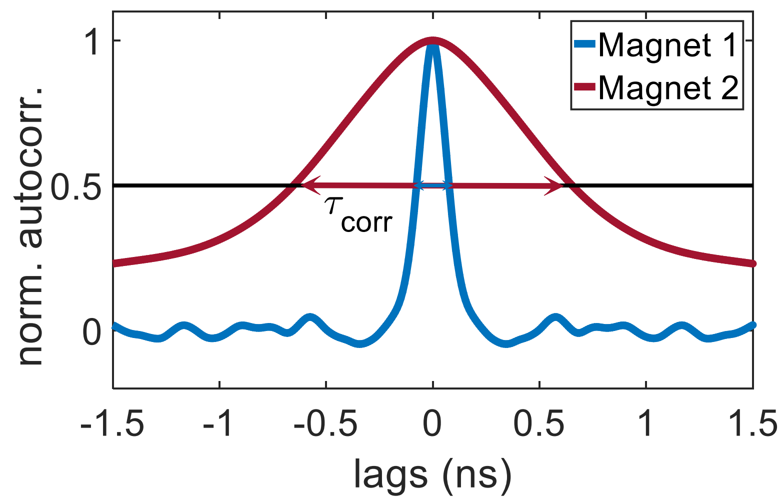
6d. Speed





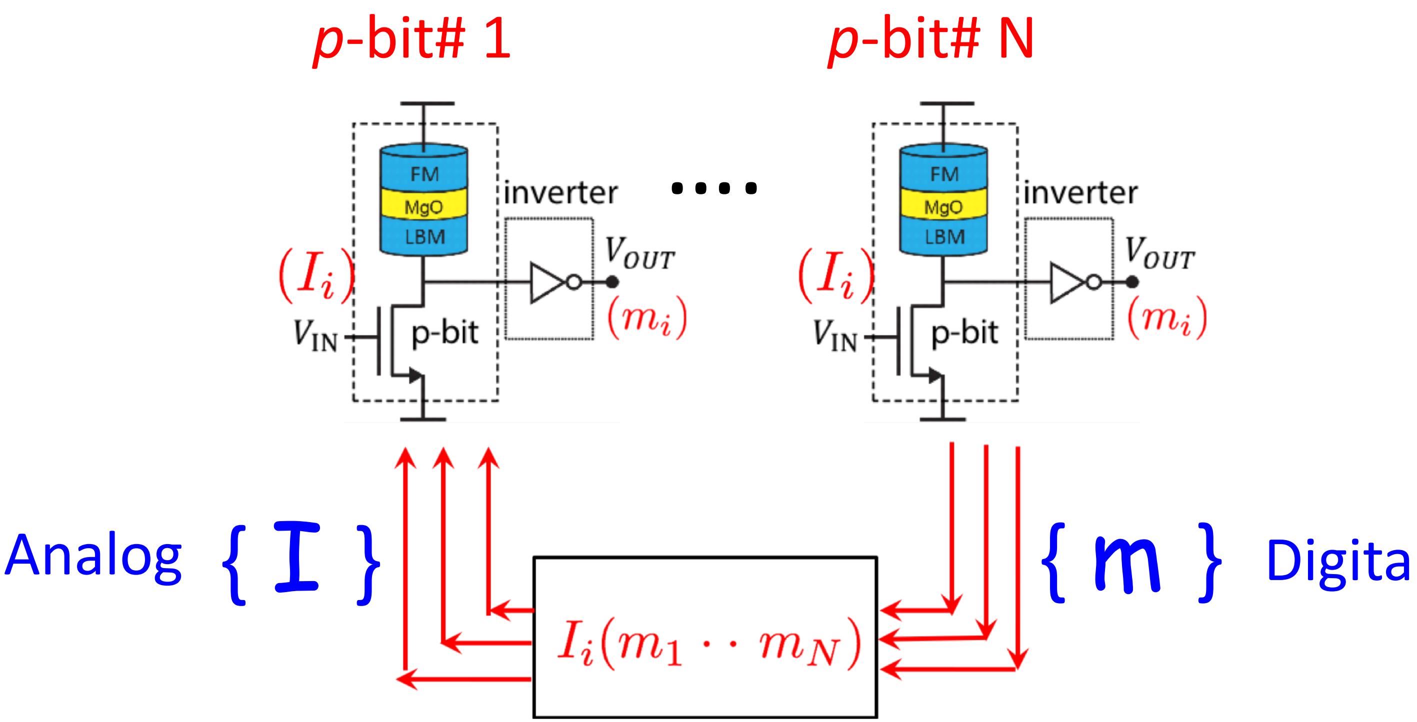
Hardware Accelerator

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



6e. *p*-Circuits

- **Reciprocal**
- **Directed**



6f. *p*-Circuits

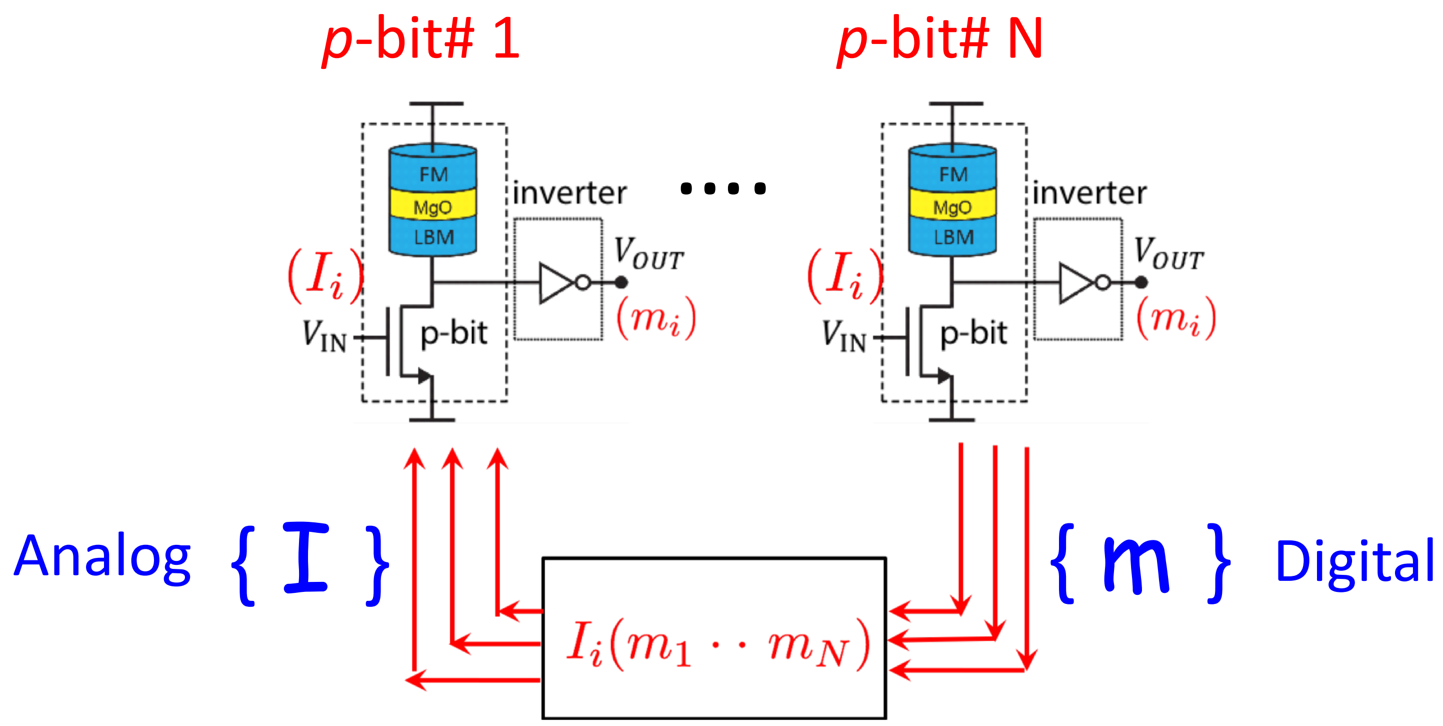
Stochastic Neural Networks

- Machine Learning
- Inference

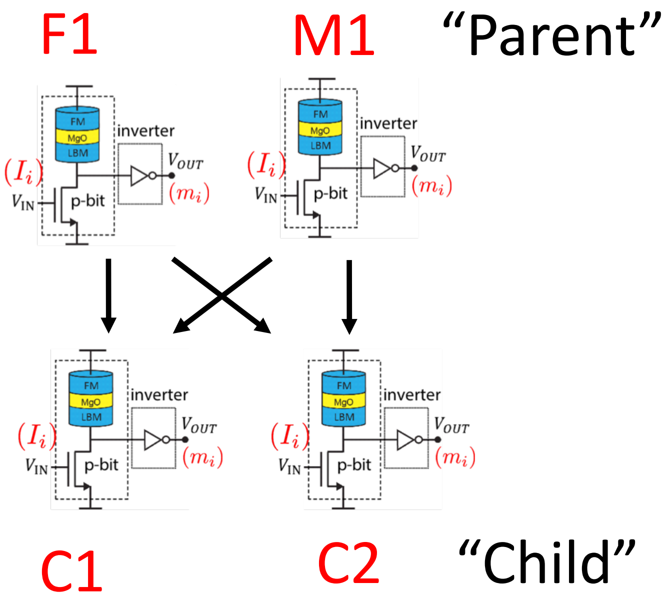
- **Reciprocal**
- **Directed**

Adiabatic Quantum Computing

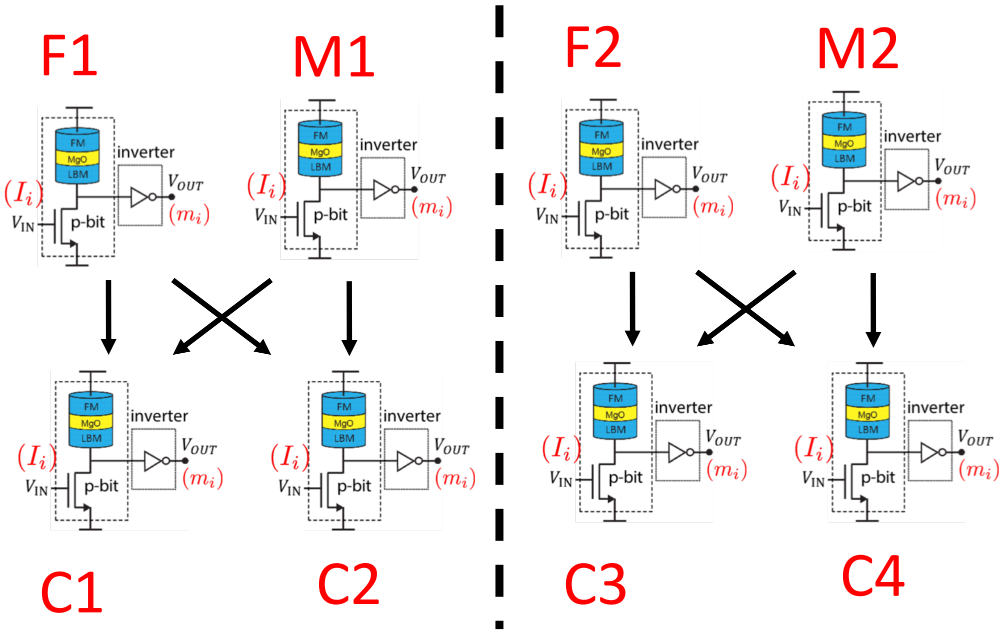
- Optimization
  - Invertible logic
  - Factorization



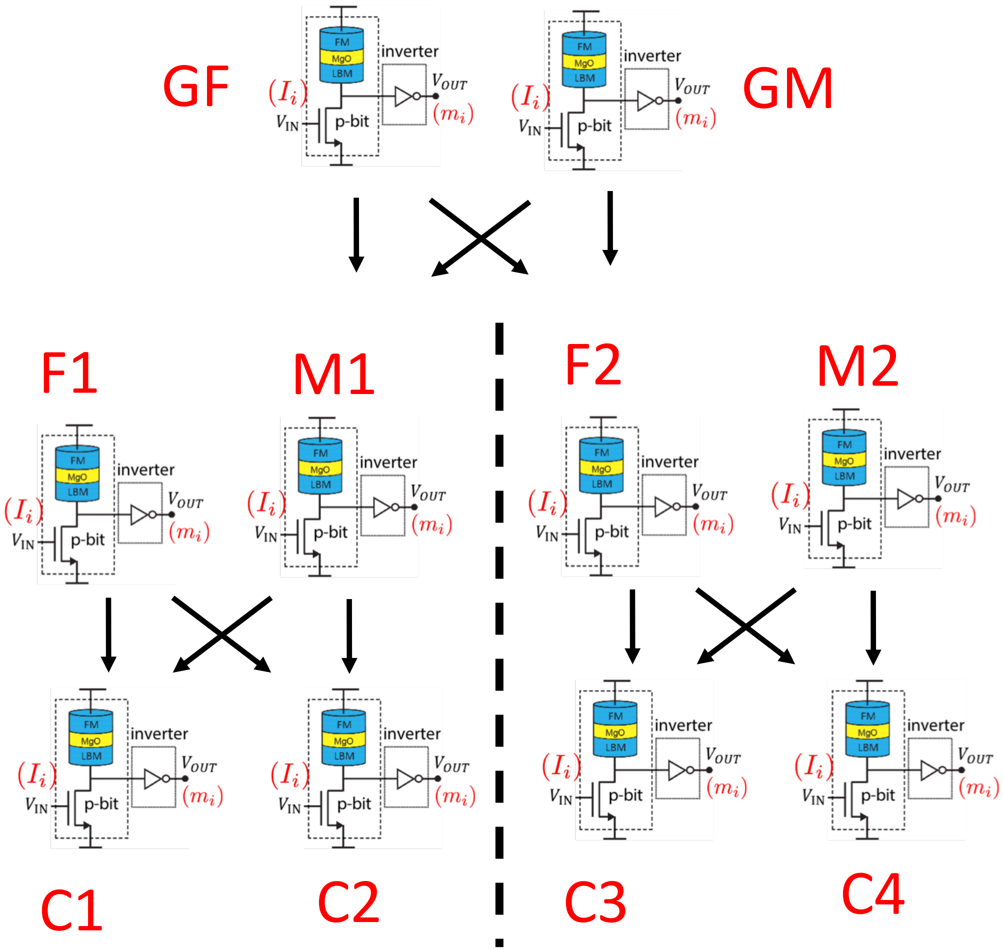
7a. A directed circuit



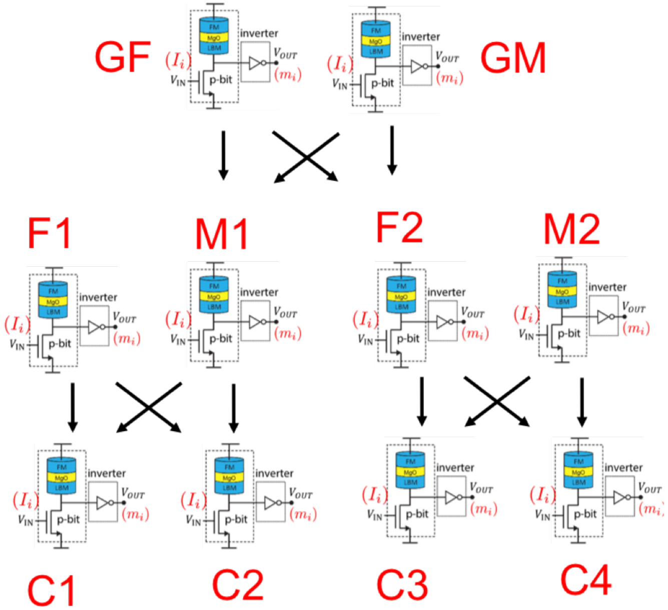
7b. A directed circuit



7c. A directed circuit



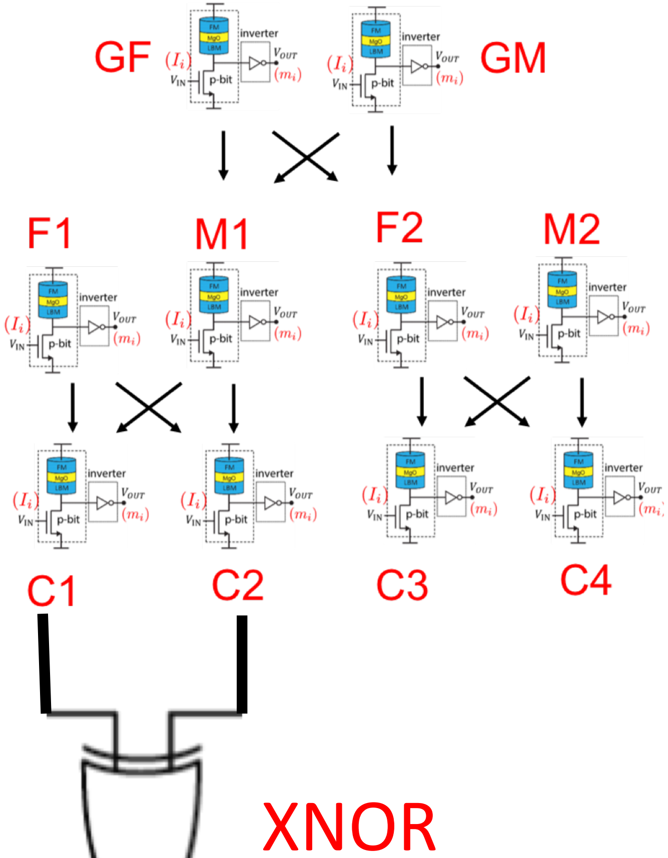
7d. A directed circuit

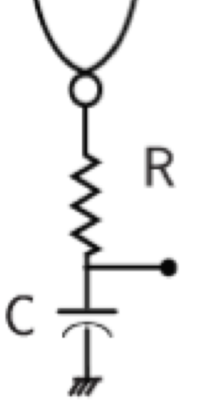


$$\langle C1 \times C2 \rangle =$$

$$\int_0^T \frac{dt}{T} \, C_1(t) C_2(t)$$

7e. A directed circuit

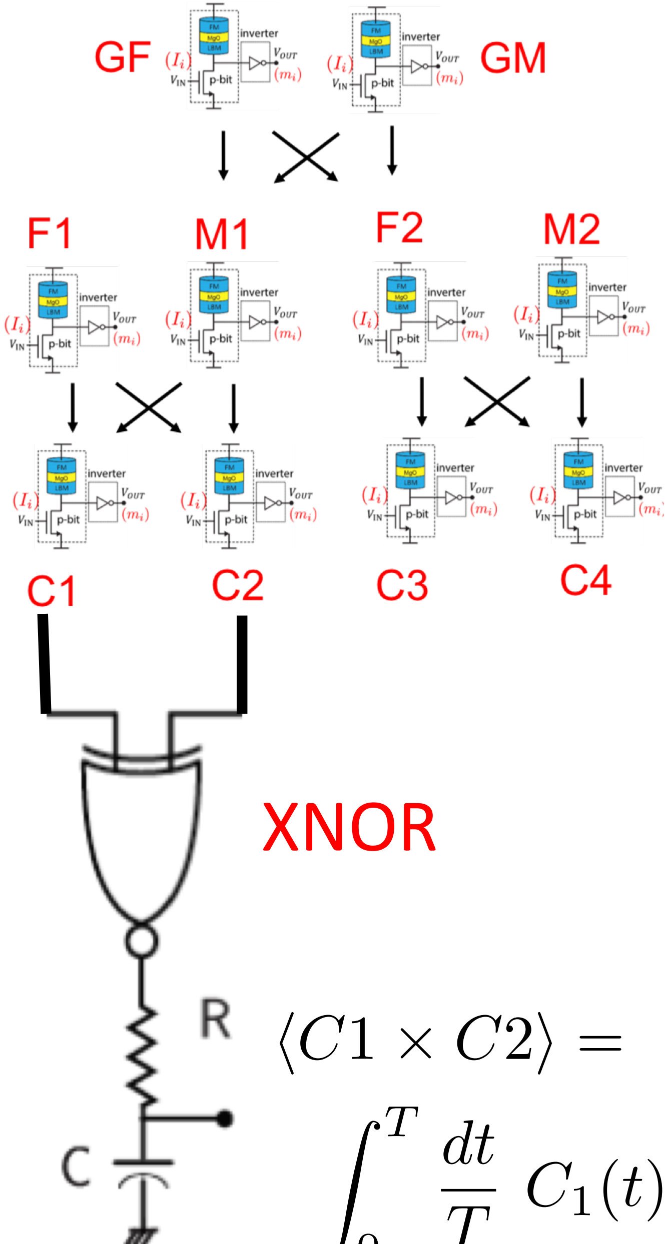




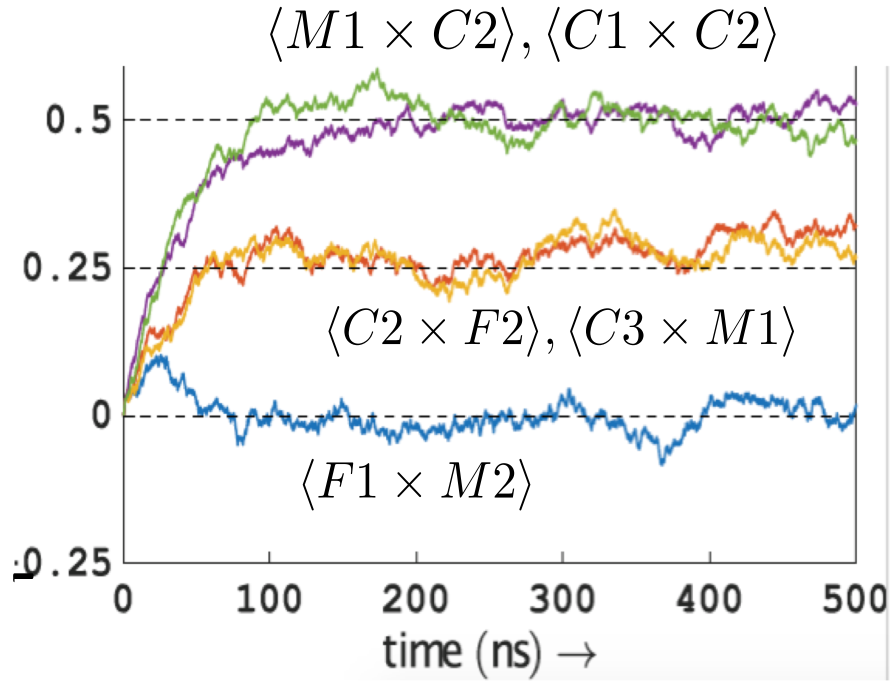
$$\langle C1 \times C2 \rangle = \int_0^T \frac{dt}{T}\, C_1(t) C_2(t)$$

A	B	XNOR
0	0	1
0	1	0
1	0	0
1	1	1

7f. A directed circuit



$$\langle C1 \times C2 \rangle = \int_0^T \frac{dt}{T}\, C_1(t) C_2(t)$$



8a. Boolean Logic

m_1	m_2	m_3	m_4	m_5	
A	B	XNOR	AND	OR	
0	0	1	0	0	4
0	1	0	0	1	9
1	0	0	0	1	17
1	1	1	1	1	31

- **Boltzmann Machine**

Boltzmann Law

$$P(m_1, \cdot\cdot, m_N) \sim e^{-E}$$

Minimum values of E for {m}'s belonging to truth table

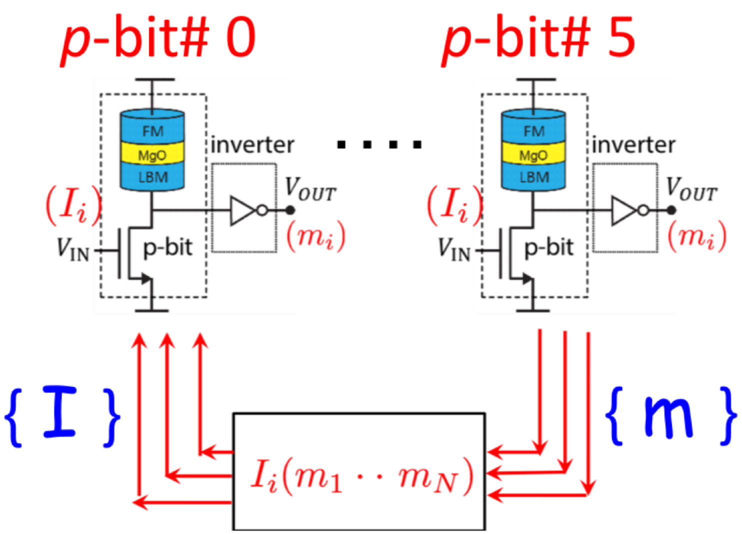
$$E = -\frac{1}{2} m^t J m$$

8b. Boolean Logic

m_1	m_2	m_3	m_4	m_5	
A	B	XNOR	AND	OR	
0	0	1	0	0	4
0	1	0	0	1	9
1	0	0	0	1	17
1	1	1	1	1	31

Minimum values of E for {m}'s belonging to truth table

$$E = -\frac{1}{2} m^t J m$$

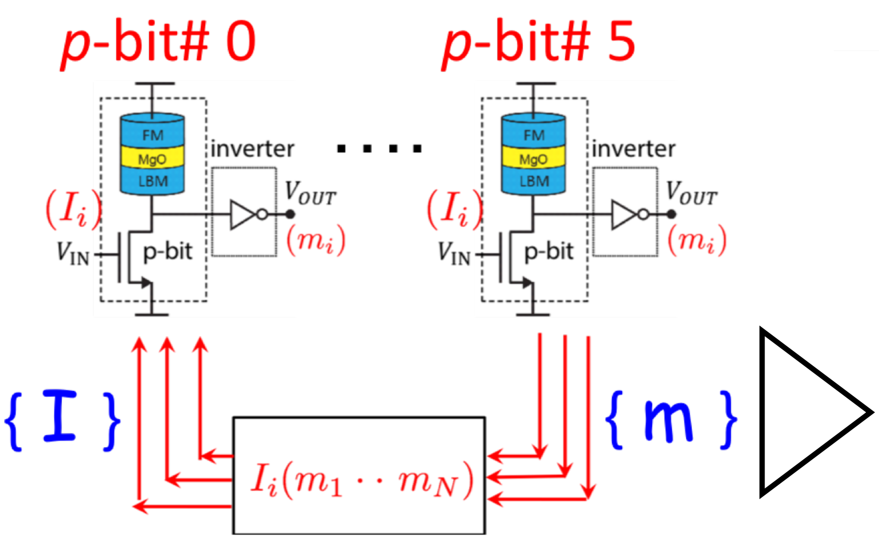


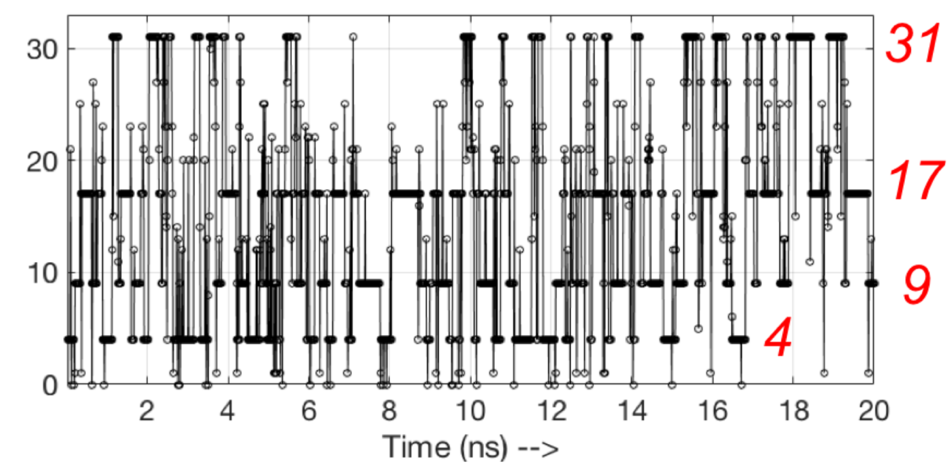
$$\begin{Bmatrix} I_0 \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & +1 & -1 & +1 \\ 0 & 0 & -1 & 0 & +1 & +1 \\ 0 & -1 & 0 & 0 & +1 & +1 \\ +1 & 0 & 0 & 0 & +1 & -1 \\ -1 & +1 & +1 & +1 & 0 & 0 \\ +1 & +1 & +1 & -1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{Bmatrix}$$

m_0 clamped to +1

8c. Boolean Logic

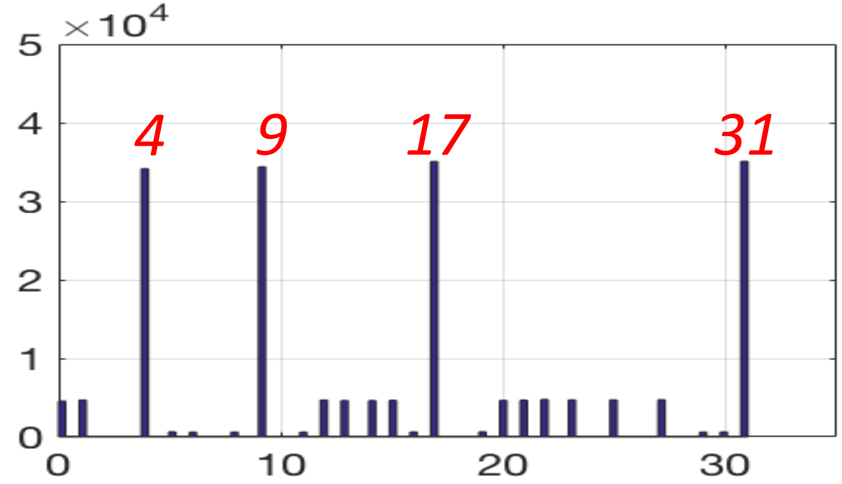
	A	B	XNOR	AND	OR
4	0	0	1	0	0
9	0	1	0	0	1
17	1	0	0	0	1
31	1	1	1	1	1

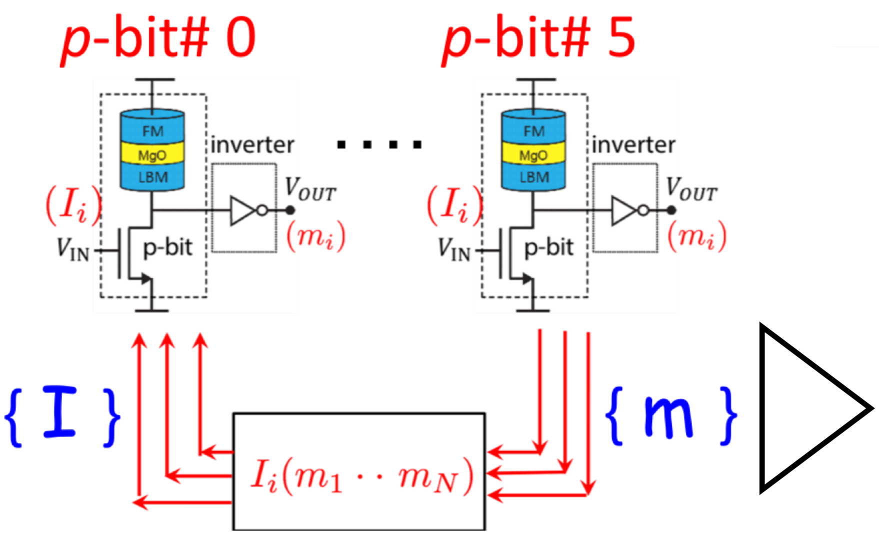


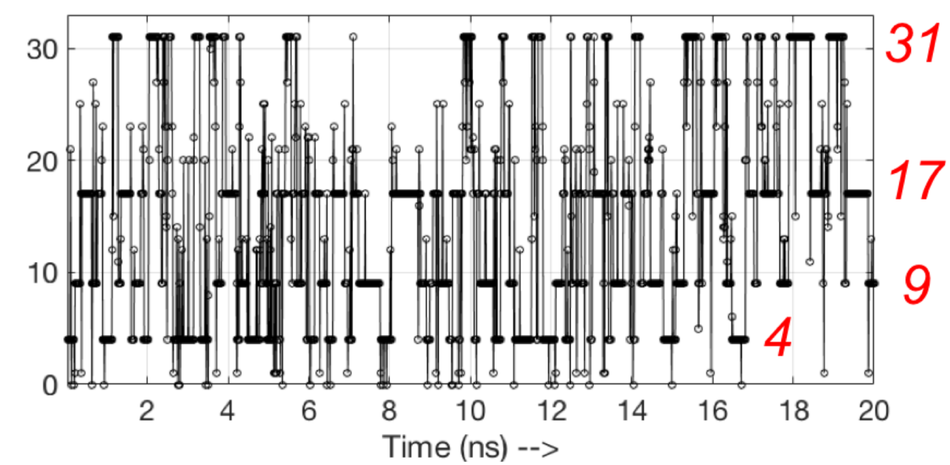


8d. Boolean Logic

	A	B	XNOR	AND	OR
4	0	0	1	0	0
9	0	1	0	0	1
17	1	0	0	0	1
31	1	1	1	1	1

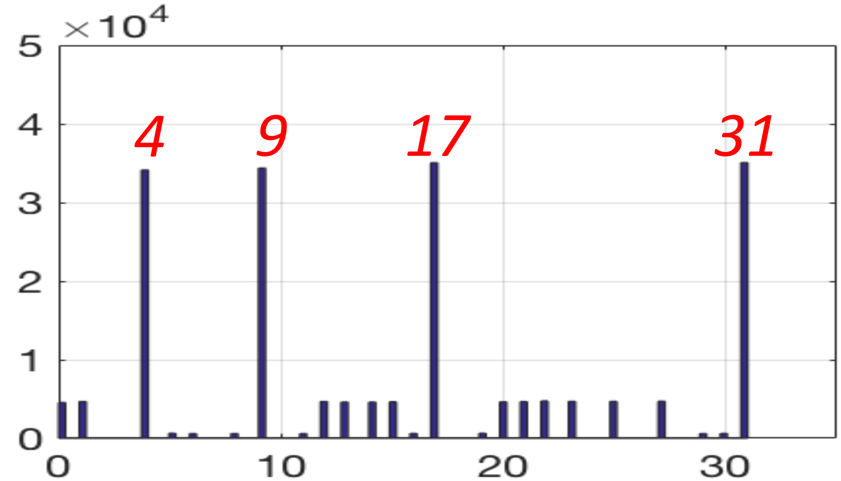




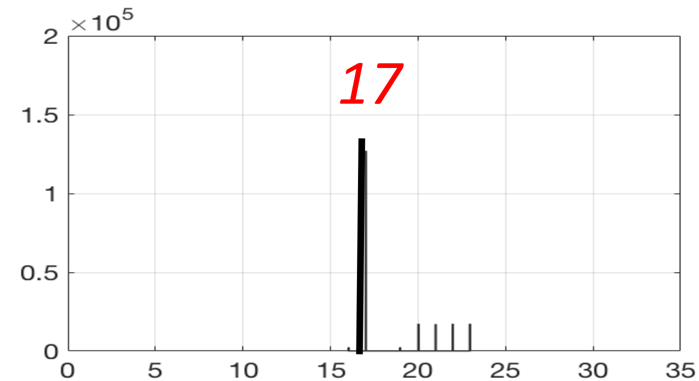


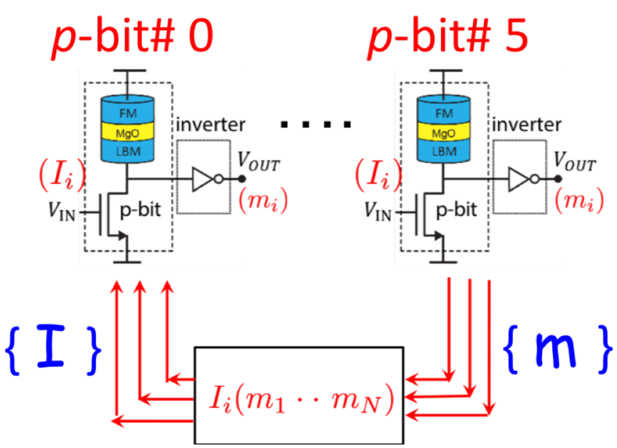
8e. Boolean Logic

	A	B	XNOR	AND	OR
4	0	0	1	0	0
9	0	1	0	0	1
17	1	0	0	0	1
31	1	1	1	1	1



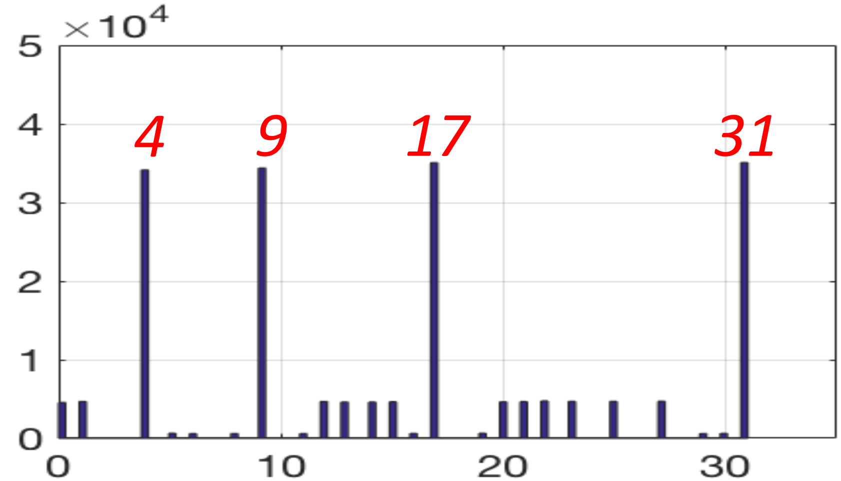
A=1
B=0

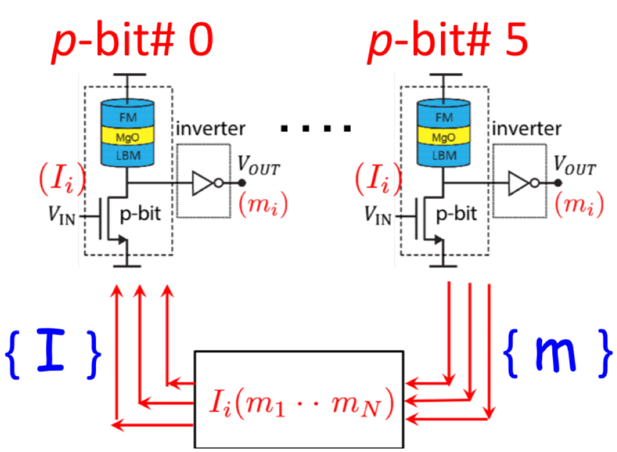




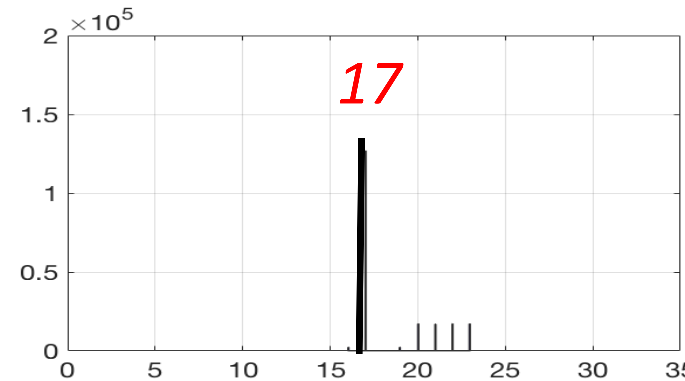
8f. Boolean Logic

	A	B	XNOR	AND	OR
4	0	0	1	0	0
9	0	1	0	0	1
17	1	0	0	0	1
31	1	1	1	1	1

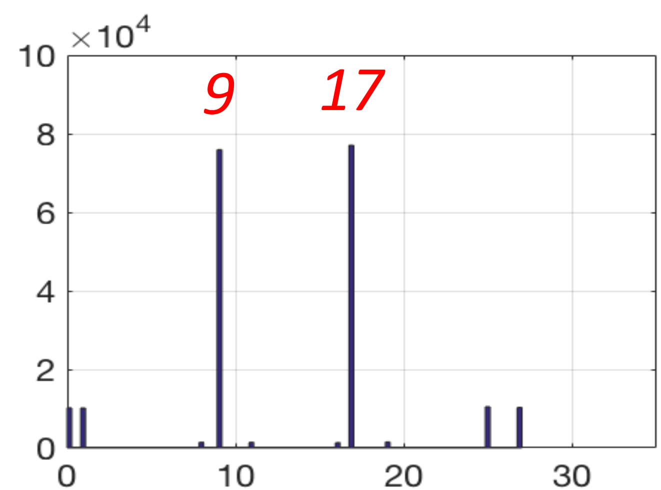




A=1
B=0

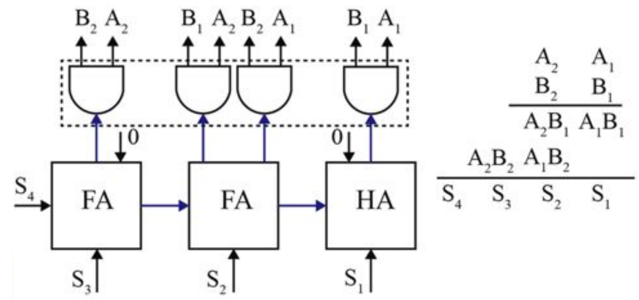


XNOR = 0

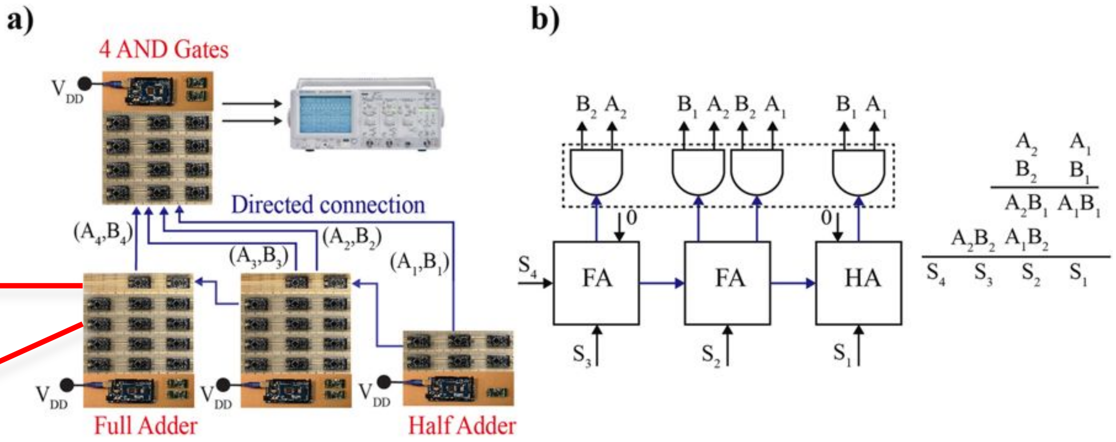


9a. Factorizer as inverse multiplier

b)



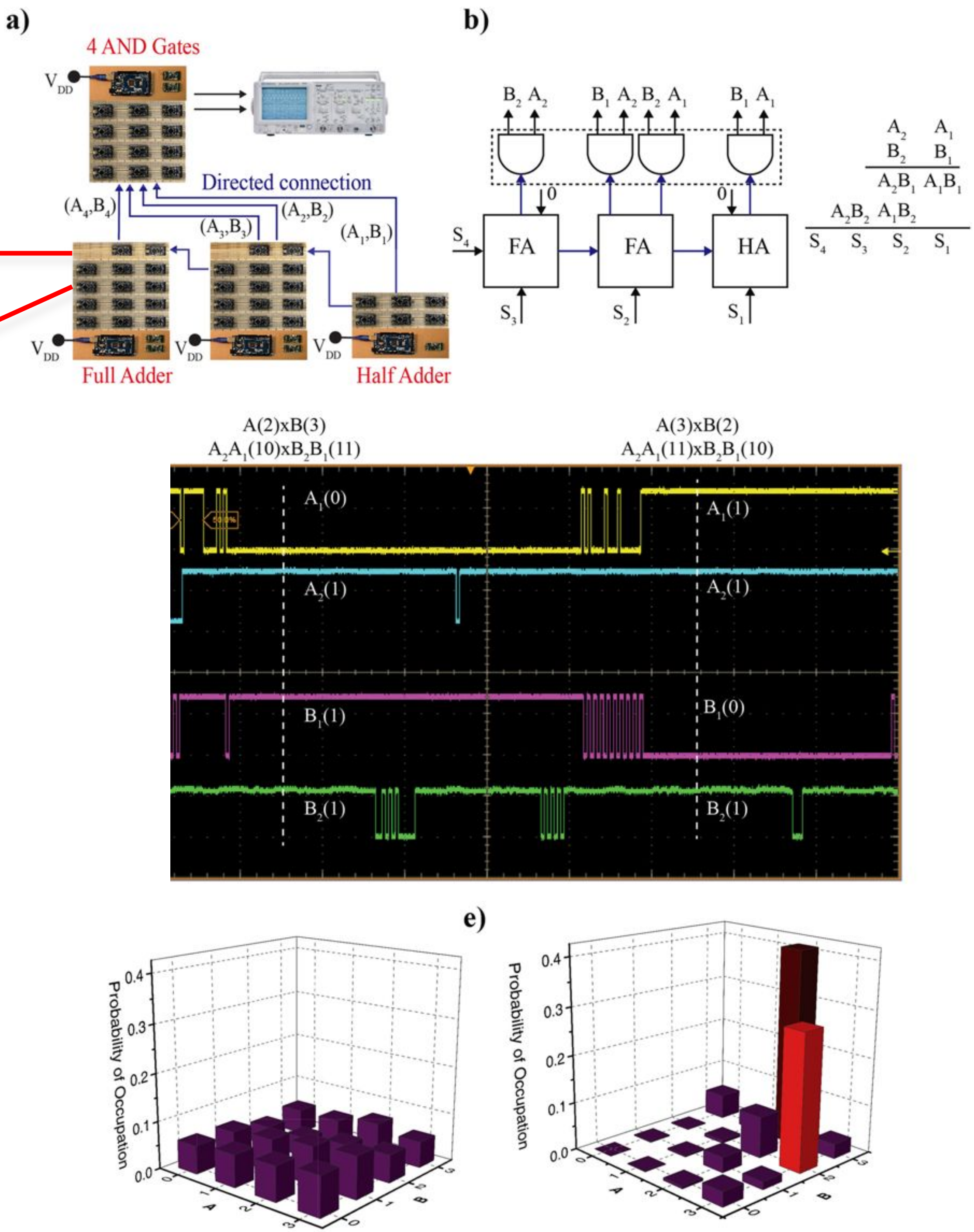
9b. Factorizer as inverse multiplier



Emulated p-bit

9c. Factorizer as inverse multiplier

Emulated p-bit



9d. Factorizer as optimizer

(p+q) p-bits

$$\boxed{x_p \,..\, x_1 \; 1}$$

$$\boxed{y_q \,..\, y_1 \; 1}$$

$$E(x_P \cdot\cdot x_1; y_Q \cdot\cdot y_1) = \left[\sum_{p=0}^{P} 2^p x_p \sum_{q=0}^{Q} 2^q y_q - N\right]^2$$

Minimum E for correct factors

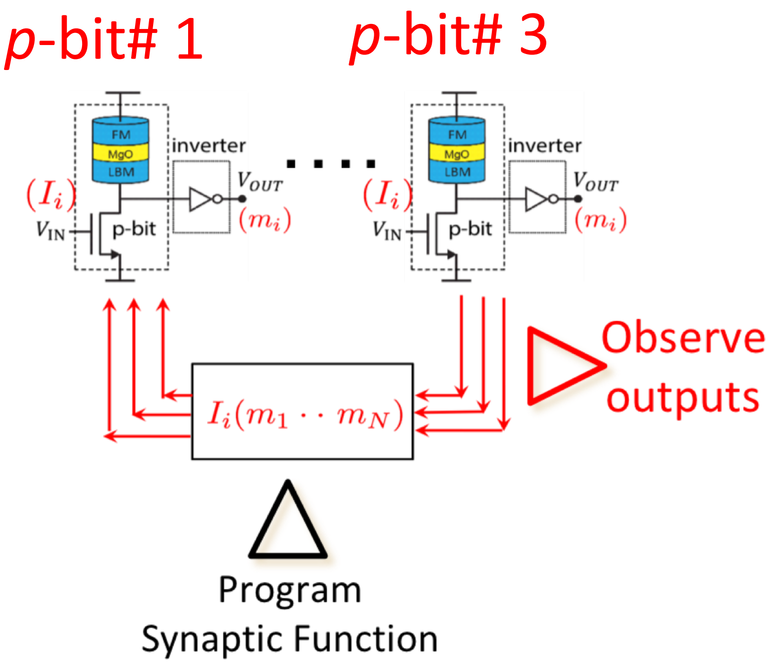
$$I_i(m_1 \cdot\cdot m_N) \quad = -\frac{\partial E(m_1, \cdot\cdot, m_N)}{\partial m_i}$$

9e. Factorizer as optimizer

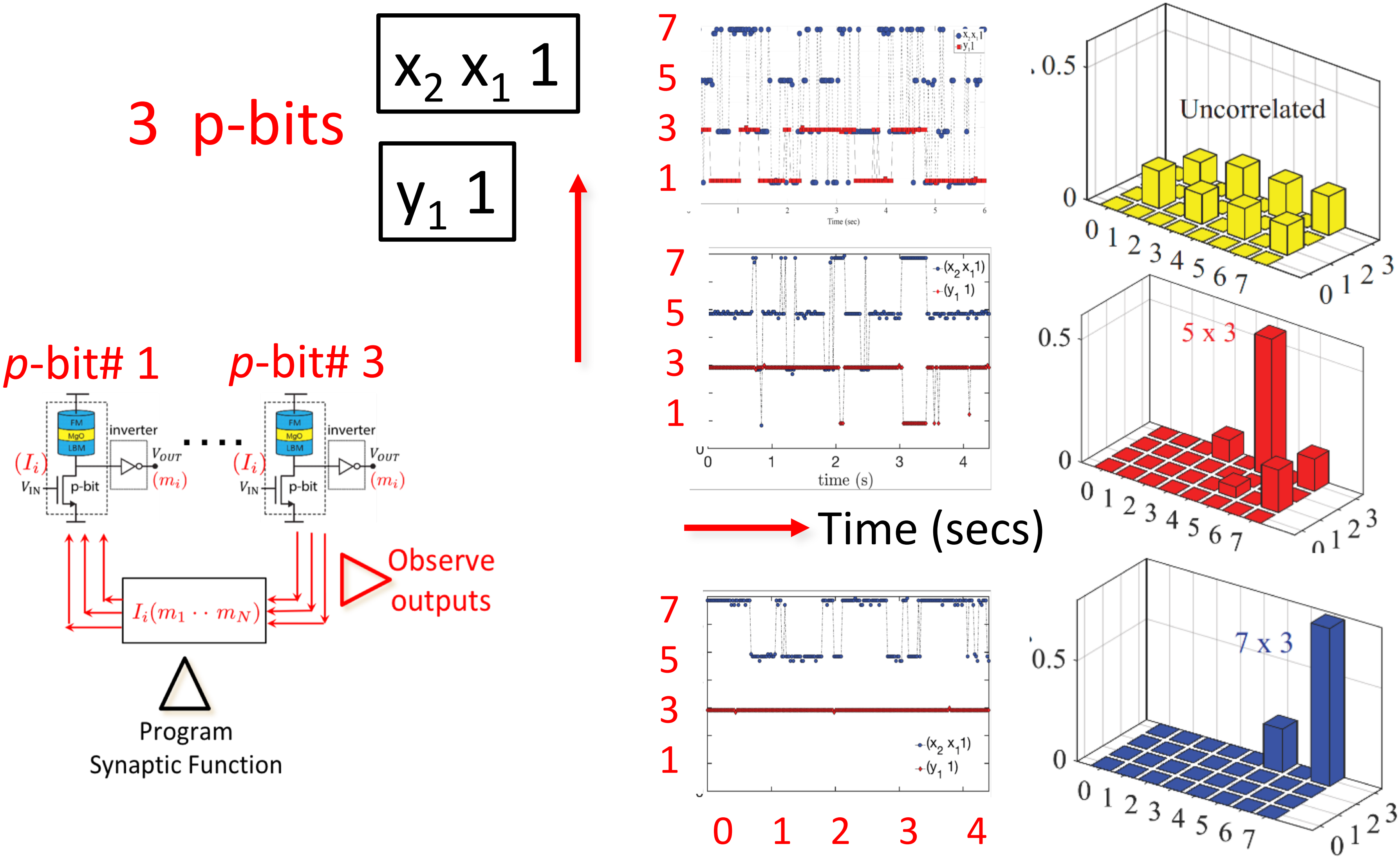
3 p-bits

$\boxed{x_2\ x_1\ 1}$

$\boxed{y_1\ 1}$



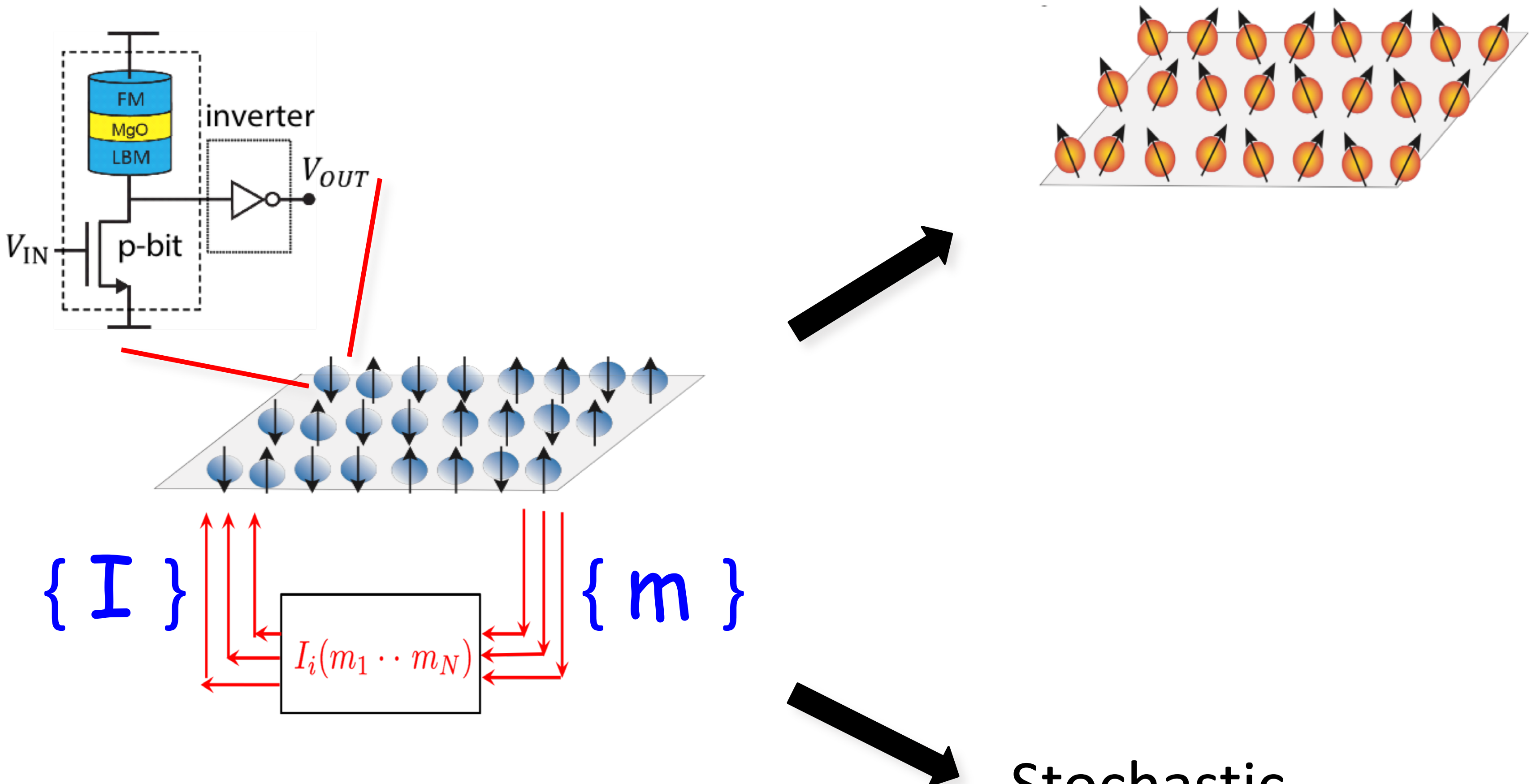
9f. Factorizer as optimizer



10a. p-circuit & q-circuit

p-circuit

q-circuit

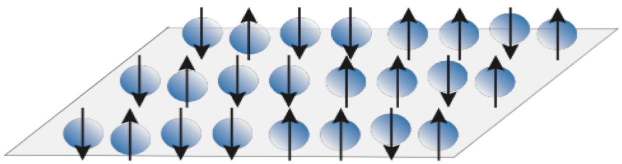


Stochastic
Neural
Networks

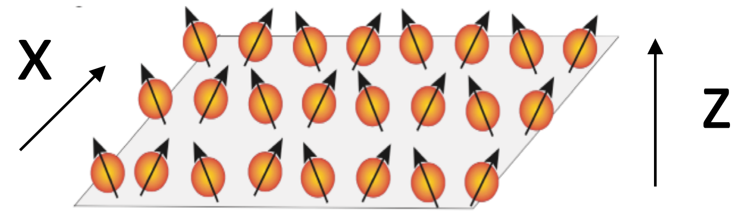
10b. p-circuit & q-circuit

p-circuit

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



q-circuit



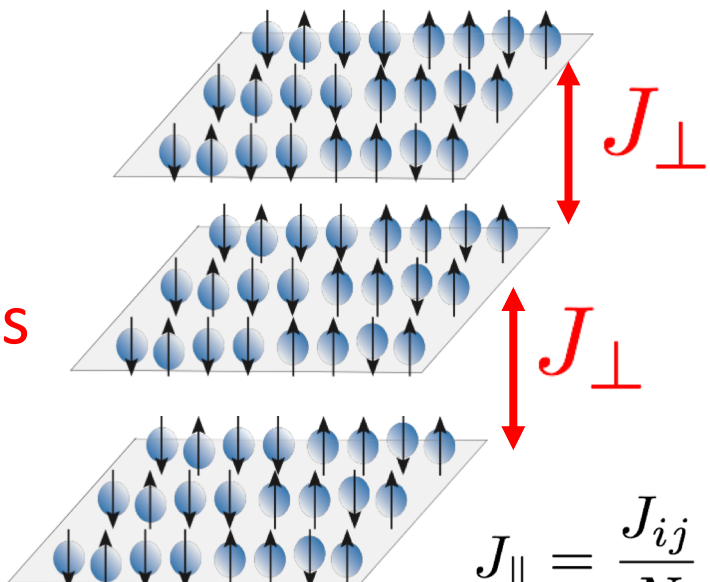
$$H = \sum_{i,j} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)} + \Gamma \sum_i \sigma_x^{(i)}$$

10c. p-circuit & q-circuit

p-circuit

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

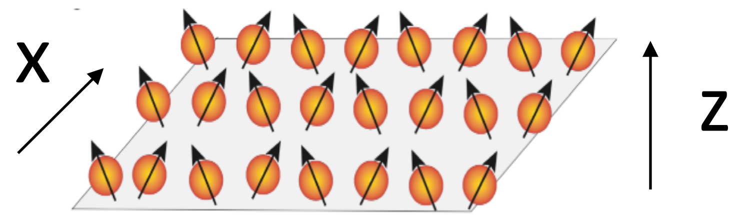
N replicas



$$J_{\parallel} = \frac{J_{ij}}{N}$$

$$J_\perp = -\frac{kT}{2} log\ tanh\left(\frac{\Gamma}{NkT}\right)$$

q-circuit



$$H = \sum_{i,j} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)} + \Gamma \sum_i \sigma_x^{(i)}$$

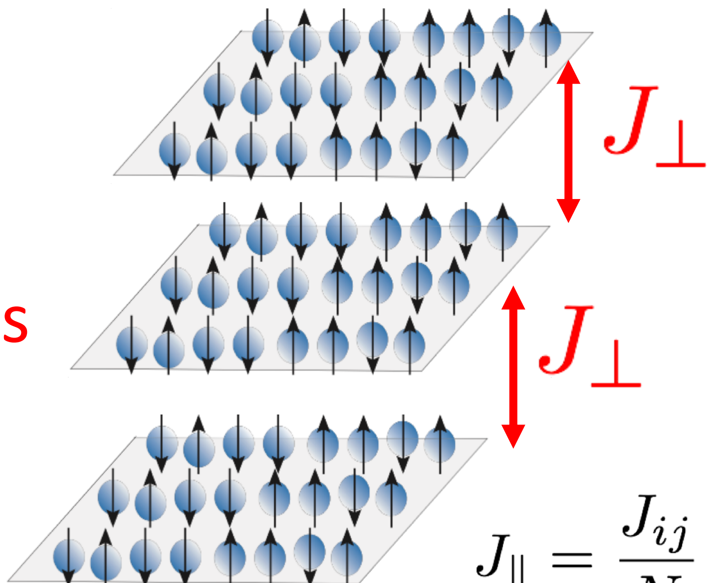
➢ Suzuki – Trotter decomposition

10d. p-circuit & q-circuit

p-circuit

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

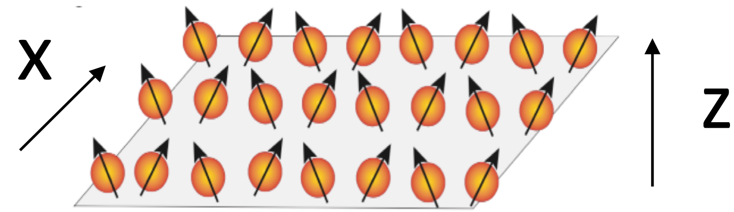
N
replicas



$$J_{\|} = \frac{J_{ij}}{N}$$

$$J_\perp = -\frac{kT}{2} log\ tanh\left(\frac{\Gamma}{NkT}\right)$$

q-circuit



$$H = \sum_{i,j} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)}$$

$$+ \Gamma \sum_i \sigma_x^{(i)}$$

Stoquastic
Hamiltonians

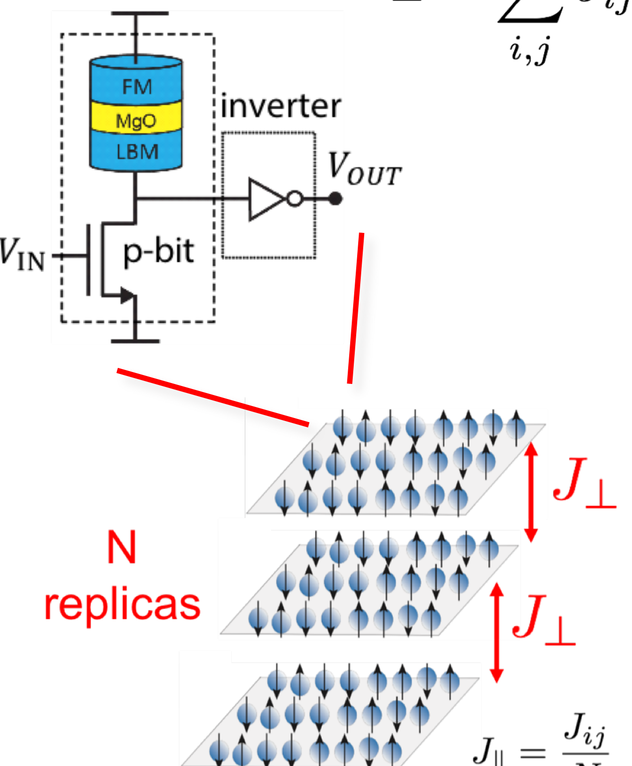
Camsari et al. (2018)
arXiv:1810.07144

➢ Suzuki – Trotter decomposition

10e. p-circuit emulating a q-circuit

p-circuit

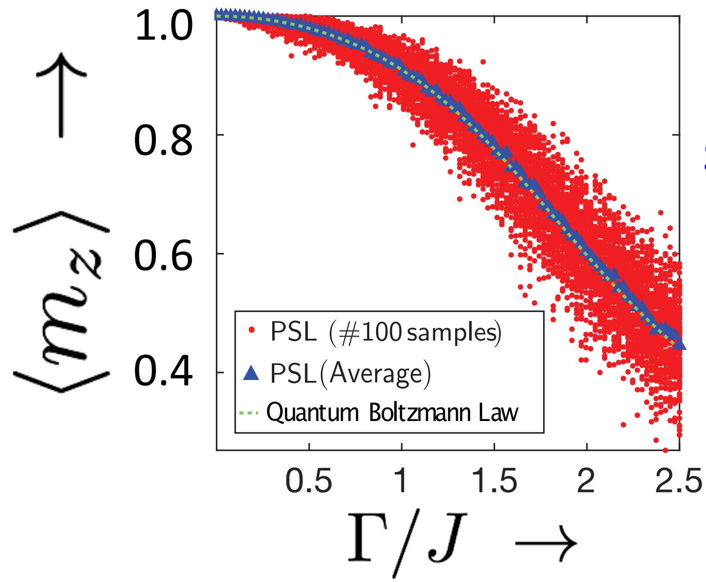
$$E = \sum_{i,j} J_{ij}\ m_i\ m_j$$



$$J_\| = \frac{J_{ij}}{N}$$

$$J_\perp = -\frac{kT}{2} log\ tanh\left(\frac{\Gamma}{NkT}\right)$$

- Suzuki – Trotter decomposition



Full SPICE simulation

Increasing Quantum Term →

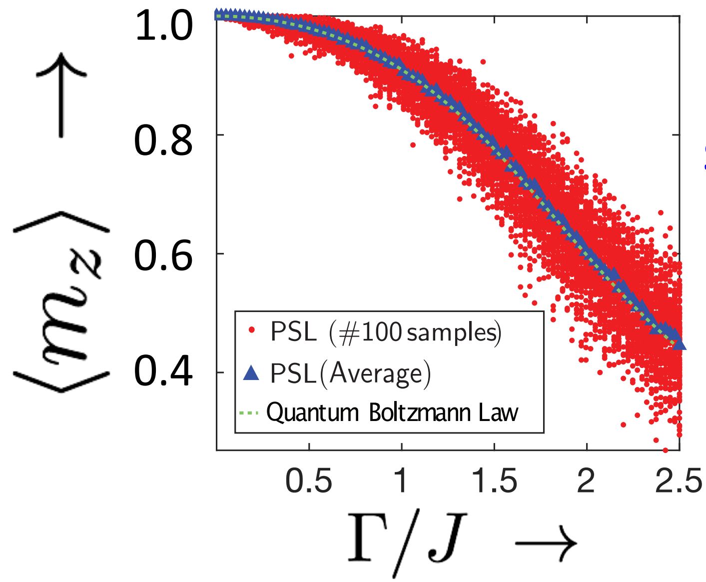
Quantum Boltzmann Law

$$\langle m_z \rangle = \frac{Trace\ [\sigma_z exp(-\beta H)]}{Trace\ [exp(-\beta H)]}$$

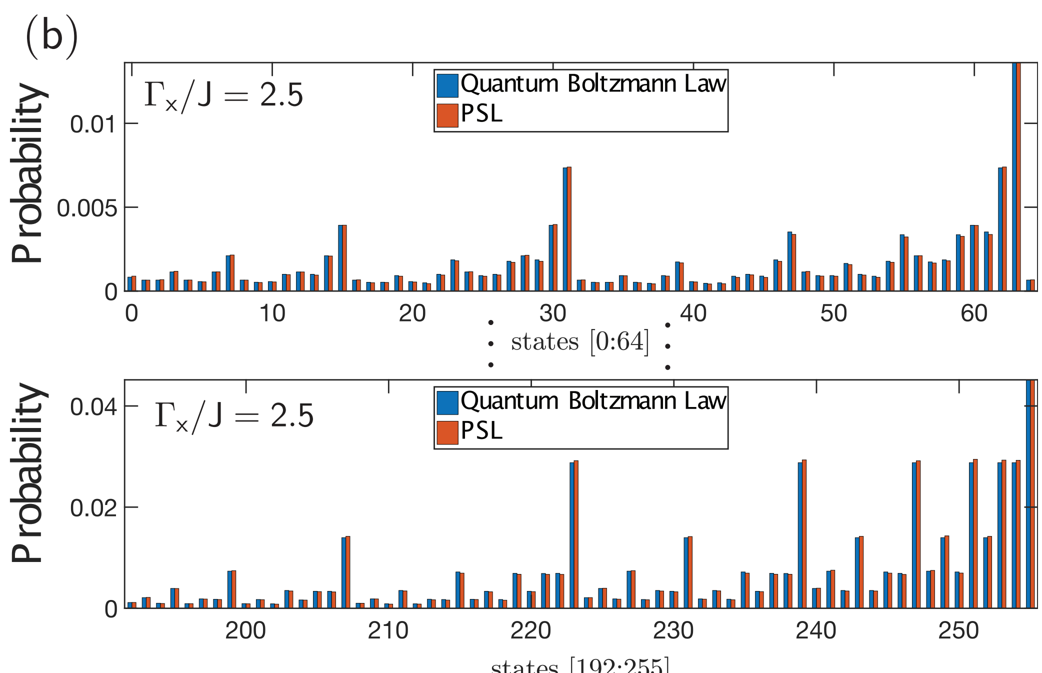
10f. p-circuit emulating a q-circuit

p-circuit

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

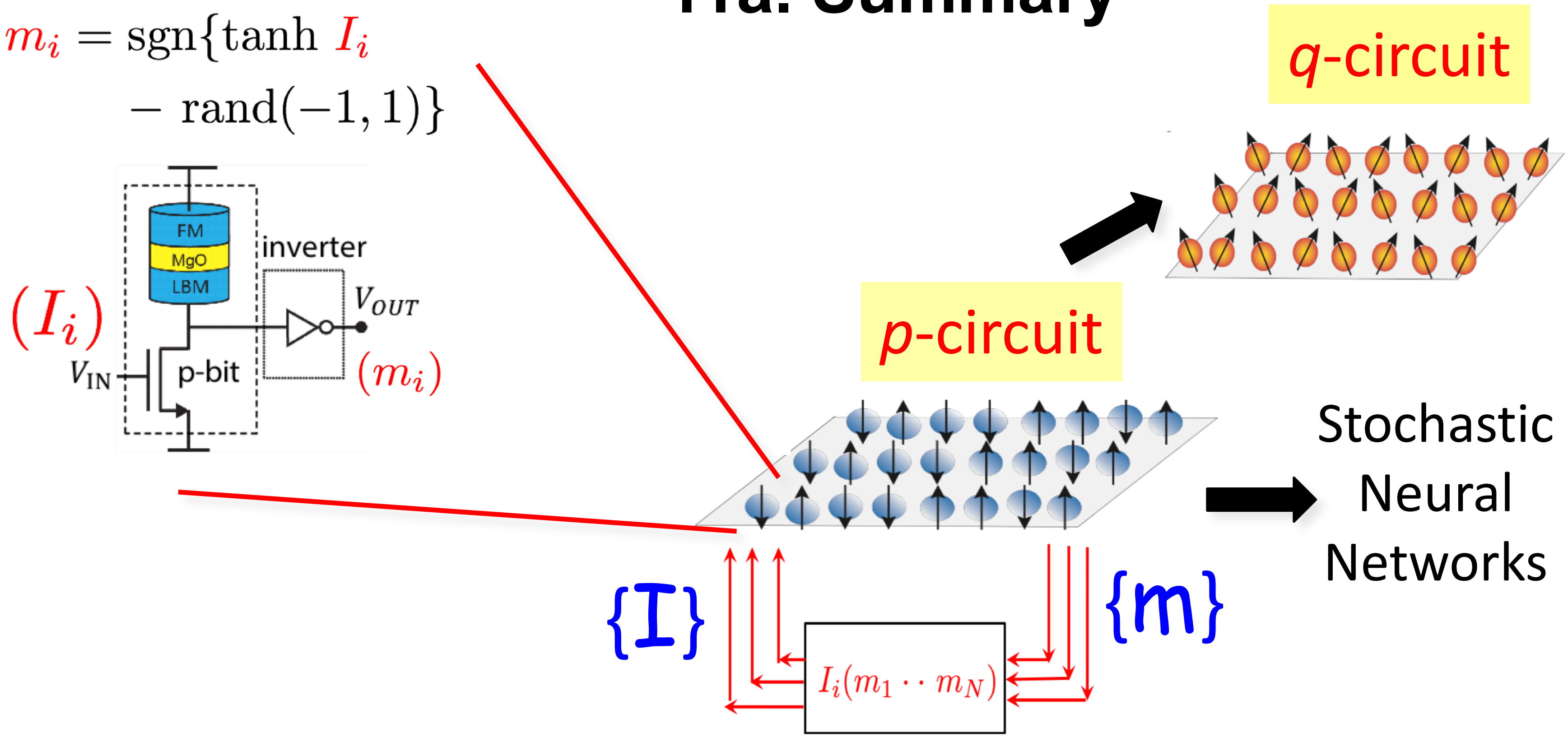


Full SPICE simulation



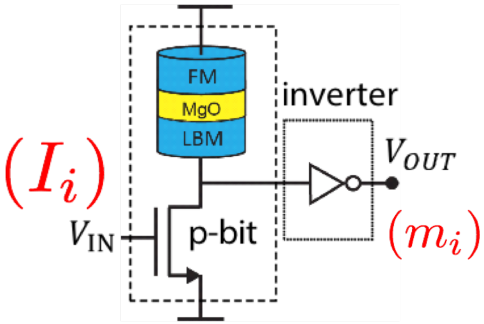
11a. Summary

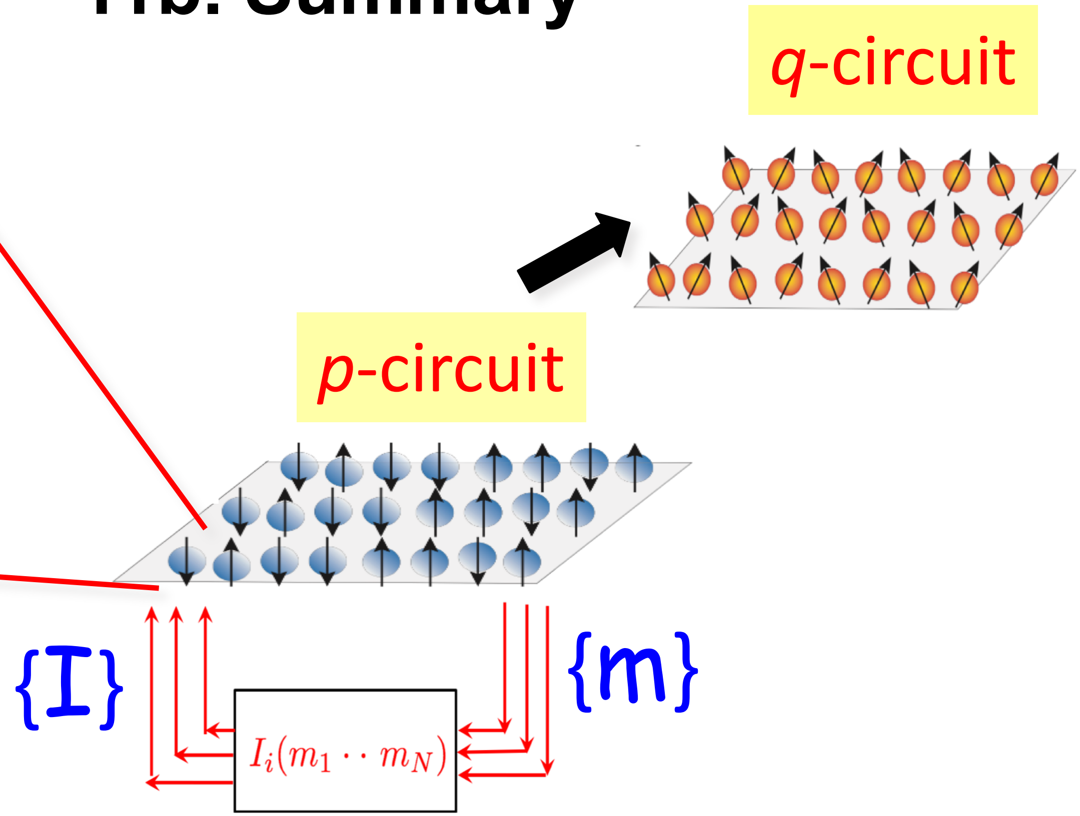
$$m_i = \mathrm{sgn}\{\tanh I_i - \mathrm{rand}(-1,1)\}$$



11b. Summary

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

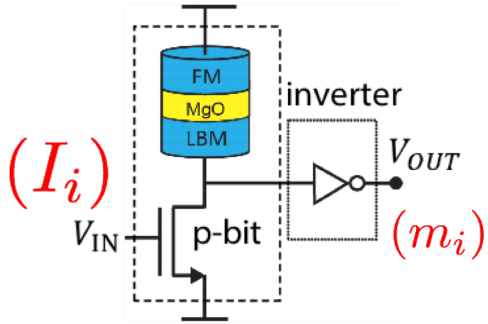


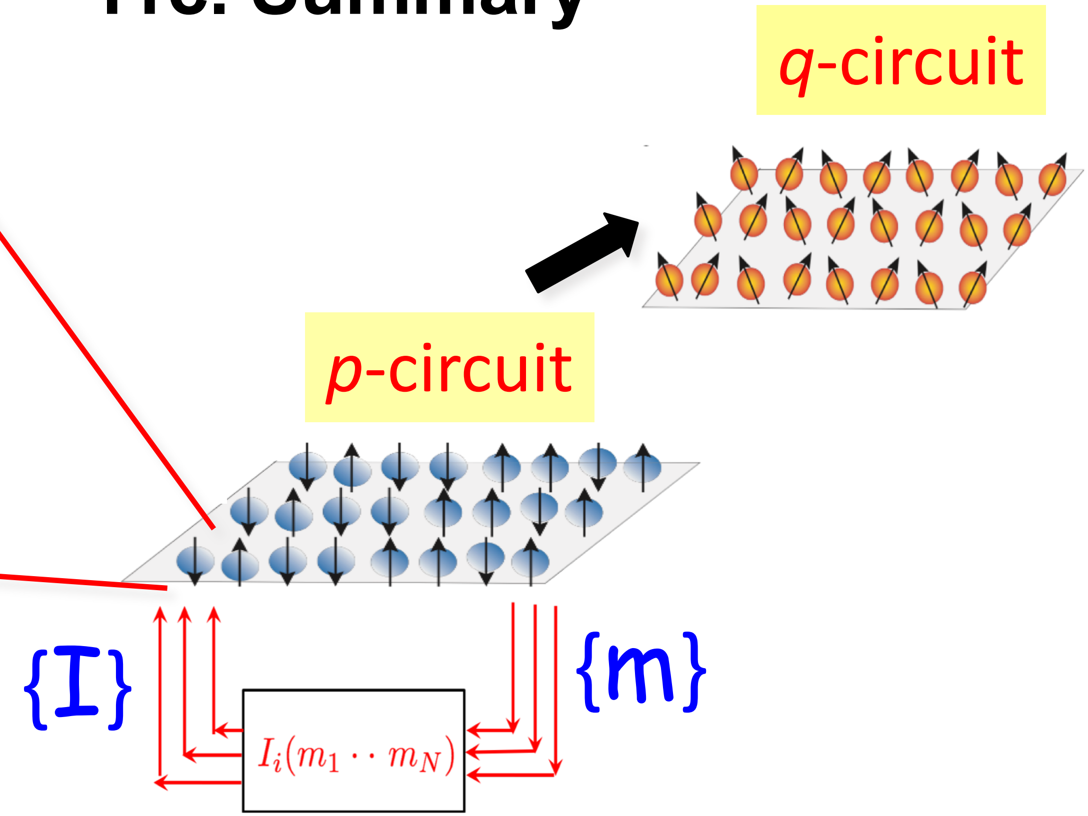


➢ Classical many-body system	➢ Quantum many-body system
Classical Interaction	**Coherent Interaction**

11c. Summary

$$m_i = \mathrm{sgn}\{\tanh I_i - \mathrm{rand}(-1,1)\}$$





➢ Classical many-body system	➢ Quantum many-body system
cf. Drift-diffusion	cf. NEGF

11d. Science Editorial

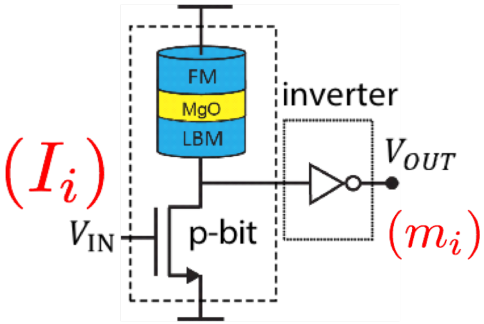
http://science.sciencemag.org/content/361/6400/313

Once a real quantum computer is realized, what's next?

- In the coming decade, we can expect that some problem-solving will be optimized much more rapidly using quantum devices.
- We can also expect that efficient sampling from a probability distribution—the theoretical version of a machine learning algorithm—will become a place where quantum computers can shine.
- In the longer term, error correction and factoring may change the landscape further.

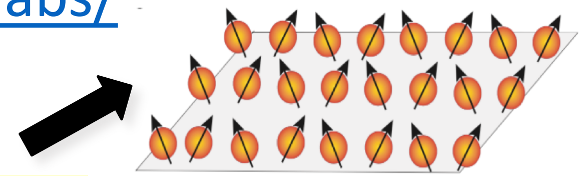
11e. Summary

$$m_i = \mathrm{sgn}\{\tanh I_i - \mathrm{rand}(-1,1)\}$$

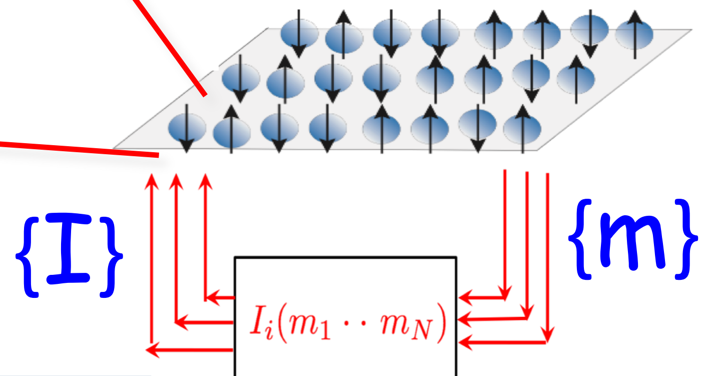


https://arxiv.org/abs/1809.04028

q-circuit



p-circuit



" Poor man's q-bit "

- Room Temperature
- Existing Technology
- Complex Interactions

➢ Classical many-body system	➢ Quantum many-body system
Classical Interaction	**Coherent Interaction**