$p$-Bits for Quantum-Inspired Algorithms

Quantum Computing

Bits $\rightarrow$ p-bits $\rightarrow$ q-bits

0 or 1 $\rightarrow$ 0 & 1

Quantum Computing

SRC
CAPSL
ASCENT
DARPA
NSF
Purdue University
Physical Representation

**Digital Computing**

- **Magnets**

**Quantum Computing**

- **Single spins**

- **bits**
  - 0 or 1

- **q-bits**
  - 0 & 1
Unstable Magnets

Digital Computing

Quantum Computing

1

Magnets

Probabilistic Computing

Unstable magnets

0 or 1

0 & 1

0 or 1

0 & 1

Single spins
“Poor Man’s q-Bit”

Digital Computing

Quantum Computing

Many algorithms are transferable

Bits (0 or 1)

P-bits

Q-bits (0 & 1)

Probabilistic Computing

• Room Temperature
• Existing Technology
"Poor Man's q-Bit"

Digital Computing

0 or 1

Quantum Computing

0 & 1

Probabilistic Computing

- Room Temperature
- Existing Technology

Different from running stochastic algorithms on a digital computer

Many algorithms are transferable
p-Circuit & q-Circuit

p-Computing

Classical Interaction

- Classical many-body system

q-Computing

Coherent Interactions

- Quantum many-body system
2^N states .. Impractical to calculate all the probabilities on a deterministic computer, if N is large

... the other way to simulate a probabilistic nature, N .. is by a computer, C .. which itself is probabilistic, .. in which the output is not a unique function of the input. ...”
Building a p-Circuit

Need

3-terminal device

p-Computing

Classical Interaction
Simple Example
Building Correlations

\[
\{V_{IN}\} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \{V_{OUT}\}
\]
More generally...

\[ \{V_{IN}\} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \{V_{OUT}\} \]
Implementing a 3-terminal $\rho$-Bit

\[ \frac{V_{\text{out}}}{v_0} = \vartheta \left[ \sigma \left( \frac{V_{\text{in}}}{v_0} \right) - r \right] \]
Fixed resistor controlled with $V_{in}$

$$R \approx R_0 \left[ 1 + \sigma \left( \frac{V_{in}}{v_0} \right) \right]$$

$$\frac{V_{out}}{v_0} = \varphi \left[ \sigma \left( \frac{V_{in}}{v_0} \right) - r \right]$$
Fluctuating resistor

$$R = R_0 (1 + r)$$

Fixed resistor controlled with $V_{in}$

$$R \approx R_0 \left[ 1 + \sigma \left( \frac{V_{in}}{v_0} \right) \right]$$

SPICE Simulation
K. Camsari, S. Salahuddin et al.
Implementing p-Bits with Embedded MTJ,
Electron Device Letters 38, 1767

\[
\frac{V_{out}}{v_0} = \mathcal{V} \left[ \sigma \left( \frac{V_{in}}{v_0} \right) - r \right]
\]
Fixed resistor controlled with $V_{in}$

$R \approx R_0 \left[ 1 + \sigma \left( \frac{V_{in}}{v_0} \right) \right]$

$R = R_0 (1 + r)$

Experiment: 3-terminal $\rho$-Bit

Fluctuating resistor

Fukami / Ohno group
Tohoku U.

$\frac{V_{out}}{v_0} = \vartheta \left[ \sigma \left( \frac{V_{in}}{v_0} \right) - r \right]$
Experiment: 3-T $\rho$-Bit

Sigmoidal fit

$V_{\text{IN}} = 1.950\text{V}$

$\langle V_{\text{OUT}}\rangle [\text{V}]$

$V_{\text{IN}} [\text{V}]$

$t [\text{ms}]$

$V_{\text{OUT}}$
Experiment: 3-T $p$-Bit
Building a $p$-Circuit
Experiment: \( p \)-Circuit

For a given \( N \) we can program the “feedback” such that states with the minimum

\[
E = (X Y - N)^2
\]

have the highest probability

\[
X = \sum_{p=0}^{p=4} 2^p x_p
\]

\[
Y = \sum_{q=0}^{q=4} 2^q y_q
\]
Experiment: $p$-Circuit

For a given $N$ we can program the “feedback” such that states with the minimum have the highest probability

\[ X = \sum_{p=0}^{p=4} 2^p x_p \]

\[ Y = \sum_{q=0}^{q=4} 2^q y_q \]

\[ E = (XY - N)^2 \]
**Experiment: p-Circuit**

Using 8 p-bits, numbers up to **945** were factorized.

\[
X = \sum_{p=0}^{p=4} 2^p x_p
\]

\[
Y = \sum_{q=0}^{q=4} 2^q y_q
\]

For a given N we can program the “feedback” such that states with the minimum

\[
E = (XY - N)^2
\]

have the highest probability.

**W.A. Borders, A.Z. Pervaiz et al.**

*Integer Factorization Using Stochastic Magnetic Tunnel Junctions,*

*Nature (in press)*
“Feedback” to minimize E

\[ \{I\} = \{V_{in}\}/v_0 \]
\[ \{m\} = \{V_{out}\}/v_0 \]

\[ I_i = -\frac{\partial E}{\partial m_i} \]

\[ X = \sum_{p=0}^{p=4} 2^p x_p \]
\[ Y = \sum_{q=0}^{q=4} 2^q y_q \]

For a given N we can program the “feedback” such that states with the minimum

\[ E = (XY - N)^2 \]

have the highest probability
Factorizing by optimization

For a given $N$ we can program the “feedback” such that states with the minimum have the highest probability.

$X = \sum_{p=0}^{p=4} 2^p x_p$

$Y = \sum_{q=0}^{q=4} 2^q y_q$

For a given $N$ we can program the “feedback” such that states with the minimum

$E = (XY - N)^2$

have the highest probability.

$\frac{\partial E}{\partial m_i}$

$p$-Computing

Algorithm from Adiabatic Quantum Computing

Difference:

Used non-linear feedback

Eliminates need for auxiliary $p$-bits
Factorizer: A Different Approach

Factorization as Optimization

Minimize

\[ E = (XY - N)^2 \]
Factorizing with an Invertible Multiplier

Can design $[W]$ so that system functions as invertible multiplier

$$\{V_{in}\} = I_0 \ [W] \{V_{out}\}$$

K. Camsari et al.  
*Stochastic p-Bits for Invertible Logic*,  
*Phys. Rev. X* 7, 031014
Factorizing with an Invertible Multiplier

52 p-bits

\[ \{V_{in}\} = I_0 \ [W] \{V_{out}\} \]

\[
\begin{align*}
&\text{p-Computing} \\
&\text{\#1} \quad \text{\#2} \quad \text{\#N} \\
&v_{\text{in}} \quad \ldots \quad v_{\text{in}} \\
&\uparrow \quad \uparrow \quad \uparrow \\
&v_{\text{out}} \quad \ldots \quad v_{\text{out}} \\
&N \quad \ldots \quad N \\
&\text{\{\text{V}_{\text{in}}\}} \quad \text{\{\text{V}_{\text{out}}\}}
\end{align*}
\]

\[ N = 143 \]
\{V_{in}\} = I_0 \ [W] \{V_{out}\}

"Thermal" Annealing

Electrically controlled

N = 143

\[
I_0 = 0 \text{ to } 10
\]
Quantum Annealing

\[ l_0 = 10 \]
\[ B_x = 10 \]

\[ \rightarrow = \uparrow + \downarrow \]

\[ B_x \]

q-Computing

Coherent Interactions

\[ l_0 = 0 \text{ to } 10 \]
\[ B_x = 0 \]
Quantum Annealing

$l_0 = 10$
$B_x = 10$

$q$-Computing

Coherent Interactions

$l_0 = 10$
$B_x = 10$ to 0

$B_x$

$l_0 = 0$ to 10
$B_x = 0$

100 ensembles

100 ensembles
Quantum Annealing with $p$-Bits

$p$-Computing

$q$-Computing

Coherent Interactions

Classical Interaction

$l_0 = 10$

$B_x = 10$ to $0$

100 ensembles
Replica Method

Classical Interaction

q-Computing

Coherent Interactions

Replica Method

\[ I_0 = 10 \]
\[ B_x = 10 \text{ to } 0 \]

Suzuki-Trotter Transformation
Replica Method

Classical Interaction

p-Computing

Basis for Quantum Monte Carlo

- Can address many (but not all) Hamiltonians
  - “sign problem”
  - Some require negative probabilities ("non-stoquastic")

Suzuki-Trotter Transformation

Coherent Interactions

q-Computing
Heisenberg Hamiltonian

**Triplet**
- \( \{00\} \)
- \( \{01\} + \{10\} \)
- \( \{11\} \)

\[ P \sim |\psi|^2 \]

**Singlet**
- \( \{01\} - \{10\} \)

Coherent Interaction
Heisenberg Hamiltonian

\[ H = \begin{pmatrix} 0 & 1/2 & 1/3 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 1/3 & 1/6 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix} \]

**Triplet**
- \{00\}
- \{01\} + \{10\}
- \{11\}

\[ P \sim |\psi|^2 \]

**Singlet**
- \{01\} − \{10\}

Coherent Interaction

Classical Interaction
Heisenberg Hamiltonian with $p$-Bits

Classical Interaction

Coherent Interaction

Classical Interaction

Classical Interaction
$p$-Computing : A Bridge to $q$-Computing

- Room Temperature
- Existing Technology
- Complex Interactions
cf. Stochastic Neural Network

Commonly implemented in software

Why hardware?

Binary Stochastic Neuron

Synapse

\[ V_{\text{out}} \rightarrow \]

\[ V_{\text{in}} \rightarrow \]

\[ V_{\text{out}} \rightarrow \]

\[ V_{\text{in}} \rightarrow \]
Hardware Accelerator

Commonly implemented in software

Why hardware?

• Speed
• Power
• Area

Binary
Stochastic
Neuron

Synapse

3 transistors + s-MTJ

1000+ transistors

VIN ➔ VOUT ➔ VIN

S-MTJ
NMOS
V_{IN}
R_{SOURCE}
V_{REF}
V_{IN}
V_{OUT}
V_{dd}

Activation Function (LUT)

32 bit Comp

LFSR 32
1 2 22 32

V_{in} ➔ V_{out}

V_{OUT} ➔ V_{IN}
Sequential versus Autonomous

Commonly implemented in software

Why hardware?

- **Parallelism**

In software

For proper functioning,

Connected \( p \)-bits are updated in sequence

so that every \( p \)-bit updates with latest information

In hardware

All \( p \)-bits update autonomously in parallel

\[
\frac{\tau_{feedback}}{\tau_{pbit}} < 1
\]

Functions properly if synapse is fast.
Autonomous Operation

In hardware, all $p$-bits update autonomously in parallel. Functions properly if synapse is fast.

$$\frac{\tau_{feedback}}{\tau_{pbit}} < 1$$

No sequencers
Autonomous Probabilistic Computer

In hardware All $p$-bits update autonomously in parallel

No sequencers

cf. Maxwell’s Demon

“Intelligent Feedback”

Can even be designed to learn from training data

In hardware $\frac{\tau_{feedback}}{\tau_{pbit}} < 1$

Functions properly if synapse is fast.
Autonomous Probabilistic Computer

Different from running stochastic algorithms on a clocked computer

Digital Computing

Probabilistic Computing

Clockless Parallel Operation

Experimental Demonstration

Quantum Computing

0 or 1

p-bits

q-bits

0 & 1
Autonomous Probabilistic Computer

Experimental Demonstration

Digital Computing

Different from running stochastic algorithms on a clocked computer

Probabilistic Computing

Clockless Parallel Operation

- Room Temperature
- Existing Technology
- Complex Interactions

Quantum Computing

Many algorithms are transferable
Quantum Computing since Democritus
Scott Aaronson

“.. if you want a universe with certain very generic properties, you seem forced to one of three choices:

(1) determinism,
(2) classical probabilities,
(3) quantum mechanics “
Acknowledgements

Tohoku U.
Prof. Hideo Ohno

Prof. Sayeef Salahuddin
Berkeley

Dr. Behtash Behin-Aein
Global Foundries

Dr. Kerem Camsari

Ahmed Zeeshan Pervaiz

Shuvro Chowdhury

Jan Kaiser

Vaibhav Ostwal

Punya Debashis

Prof. Zhihong Chen

Prof. Joerg Appenzeller

Brian Sutton

Rafatul Faria

Orchi Hassan

Anirudh Ghantasala

PURDUE

Public website allowing users to explore p-bit networks