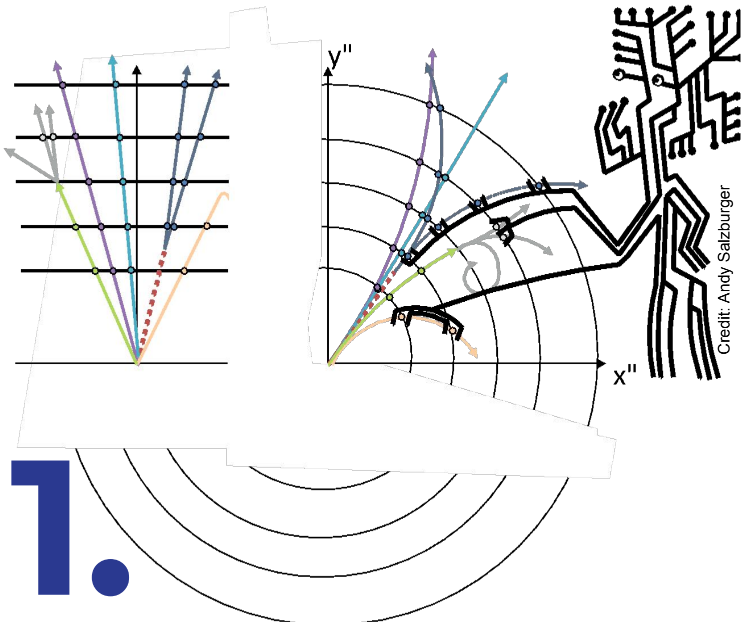
Quantum Pattern Recognition for High-Luminosity Era

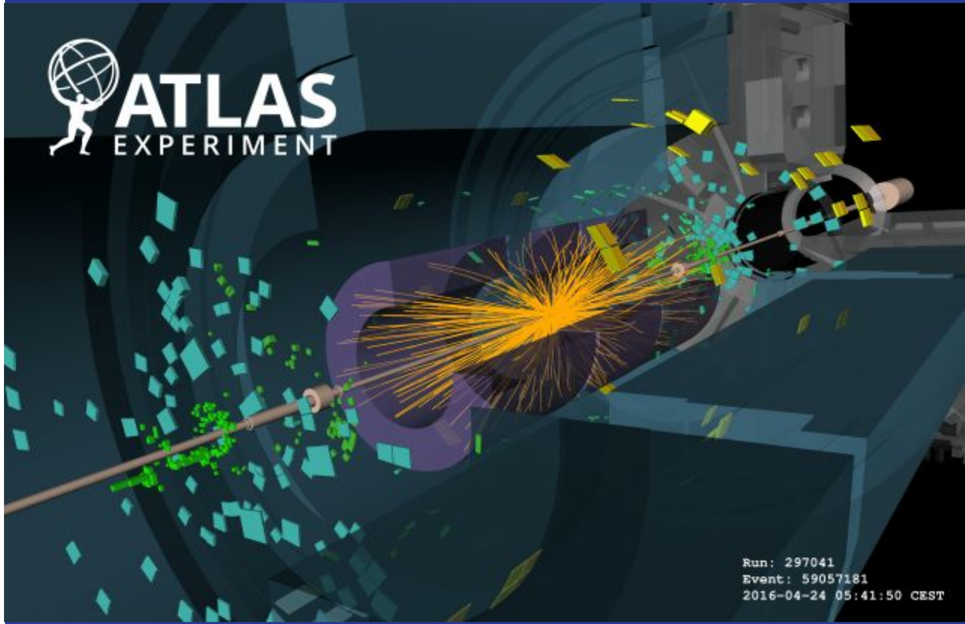
Illya Shapoval, Paolo Calafiura
Lawrence Berkeley National Laboratory

Connecting The Dots 2018, @ University of Washington, Seattle
22 March, 2018



1.

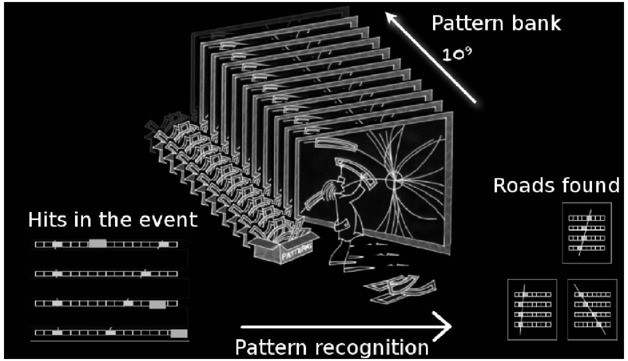
ATLAS Real-time Pattern Recognition



ATLAS Fast Tracker (FTK)

LHC Run 2 (2015) - Run 3 (2023)

A HARDWARE FOR REAL-TIME GLOBAL TRACK FINDING

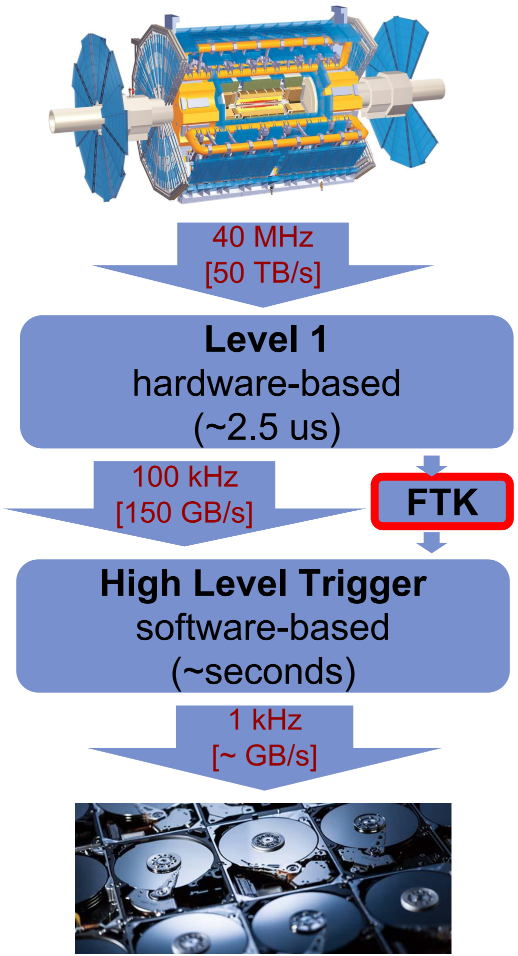


Requirements:

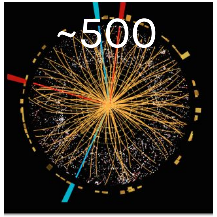
- Input: 10^8 channels
- Latency: ~100 us
- Frequency: @100 kHz

Pattern recognition engine: **Associative Memory**

- Storage: $8 \cdot 10^3$ AM custom ASIC chips
- Power: ~32 kW (+ cooling)
- Capacity: **10^9** track patterns
- Latency: average ~50 us, max ~180 us

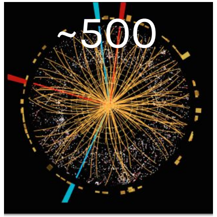


Scalability of Associative Memory

Experiment	LHC Run 2-3
LHC Luminosity ($cm^{-2}s^{-1}$)	~10^{34}
Tracks/event	
AM Capacity* (patterns)	10^{9}
AM Storage* (AM chips)	$8 \cdot 10^{3}$
Density* (patterns/chip)	128k (65 nm)

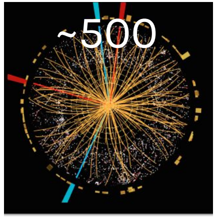
* Required by ATLAS physics and detector granularity

Scalability of Associative Memory

Experiment	LHC Run 2-3	HL-LHC (2026)
LHC Luminosity ($cm^{-2}s^{-1}$)	$\sim 10^{34}$	$\sim 10^{35}$
Tracks/event		
AM Capacity* (patterns)	10^9	$[8 - 16] \cdot 10^9$
AM Storage* (AM chips)	$8 \cdot 10^3$	$[2 - 4] \cdot 8 \cdot 10^3$
Density* (patterns/chip)	128k (65 nm)	~512k (28 nm)

* Required by ATLAS physics and detector granularity

Scalability of Associative Memory

Experiment	LHC Run 2-3	HL-LHC (2026)	HE-LHC (2030s)
LHC Luminosity (cm^{-2}s^{-1})	~10^{34}	~10^{35}	~10^{36}
Tracks/event			~50,000
AM Capacity* (patterns)	10^{9}	$[8 - 16] \cdot 10^{9}$	**?**
AM Storage* (AM chips)	$8 \cdot 10^{3}$	$[2 - 4] \cdot 8 \cdot 10^{3}$	**?**
Density* (patterns/chip)	128k (65 nm)	~512k (28 nm)	**?**

* Required by ATLAS physics and detector granularity

Memory in search applications

Memory in search applications

- Location-addressable memory
  - Pattern capacity: ***O(N/n)***, where *N* is the total number of **bits**, and **n** - the pattern length
  - Slow recall (primitive cells and high address/word handling impedance)
  - Low cost and low power dissipation

Memory in search applications

- Location-addressable memory
  - Pattern capacity: ***O(N/n)***, where *N* is the total number of **bits**, and **n** - the pattern length
  - Slow recall (primitive cells and high address/word handling impedance)
  - Low cost and low power dissipation
- Associative memory (a.k.a content-addressable memory)
  - Pattern capacity: ***O(N/n)*** in classical schemes
    - Hopfield networks scale as ***O(N)*** ($m \leq kN$, where $0.15 \leq k \leq 0.5$)
  - Fast recall (*O(1)* cycle ops) (cells with dedicated compare/combine circuits)
  - High cost and high power dissipation

Memory in search applications

- Location-addressable memory
  - Pattern capacity: ***O(N/n)***, where *N* is the total number of **bits**, and **n** - the pattern length
  - Slow recall (primitive cells and high address/word handling impedance)
  - Low cost and low power dissipation
- Associative memory (a.k.a content-addressable memory)
  - Pattern capacity: ***O(N/n)*** in classical schemes
    - Hopfield networks scale as ***O(N)*** (*m*≤*kN*, where 0.15≤k≤0.5)
  - Fast recall (*O(1)* cycle ops) (cells with dedicated compare/combine circuits)
  - High cost and high power dissipation
- Quantum associative memory
  - Pattern capacity: ***O(2^N)***, where *N* is the total number of **qubits**, and **n** - the pattern length
  - Recall time needs evaluation, high volatility with hardware technology
  - Market costs are far from "ground state" yet, relaxation time is ~10-15 years

2.

Quantum Associative Memory

Quantum Memory

- Represent pattern $\xi^i \equiv (\xi_1, \xi_2, \ldots, \xi_d)$ by a **basis state** in the Hilbert space of *d* quantum information units:

$$|\xi^i\rangle \equiv |\xi_1, \xi_2, \ldots, \xi_d\rangle$$

- Represent Ξ - a set of *N* patterns - as **superposition** of the basis states:

$$|\Xi\rangle = \sum_1^N \alpha_i |\xi^i\rangle, \qquad \alpha_i \in \mathbb{C} \ \wedge \ \sum_1^N |\alpha_i|^2 = 1$$

QuAM Capacity

QuAM features exponential storage capacity of 2^d and requires $2(d+1)$[1] qubits to operate [2].

Length of detector hit identifier (bits)	8	16	32
Length of binary track pattern (bits) [3]	64	128	256
QuAM register (qubits)	130	258	514
QuAM capacity (patterns)	**$\sim 10^{19}$**	**$\sim 10^{38}$**	**$\sim 10^{77}$**

[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)
[2] *d* is the pattern length
[3] 8 logical layers of the Inner Tracker

QuAM Capacity

QuAM features exponential storage capacity of 2^d and requires $2(d+1)$[1] qubits to operate [2].

Length of detector hit identifier (bits)	8	16	32
Length of binary track pattern (bits) [3]	64	128	256
QuAM register (qubits)	130	258	514
QuAM capacity (patterns)	**$\sim 10^{19}$**	**$\sim 10^{38}$**	**$\sim 10^{77}$**

[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)
[2] *d* is the pattern length
[3] 8 logical layers of the Inner Tracker

QuAM storage protocol

A quantum circuit implementing iterative part of the storage protocol [1].

$$1.|\psi_0^1\rangle = |p_1^1, ..., p_d^1; 01; 0_1, ..., 0_d\rangle$$

$$2.|\psi_1^i\rangle = \prod_{j=1}^{d} {}^{2c}\hat{X}_{p_j^i u_2 m_j}\,|\psi_0^i\rangle$$

$$3.|\psi_2^i\rangle = \prod_{j=1}^{d} \hat{X}_{m_j}\,{}^{1c}\hat{X}_{p_j^i m_j}\,|\psi_1^i\rangle$$

$$4.|\psi_3^i\rangle = {}^{dc}\hat{X}_{m_1 \ldots m_d u_1}\,|\psi_2^i\rangle$$

$$5.|\psi_4^i\rangle = {}^{1c}\hat{S}_{u_1 u_2}(p+1-i)|\psi_3^i\rangle$$

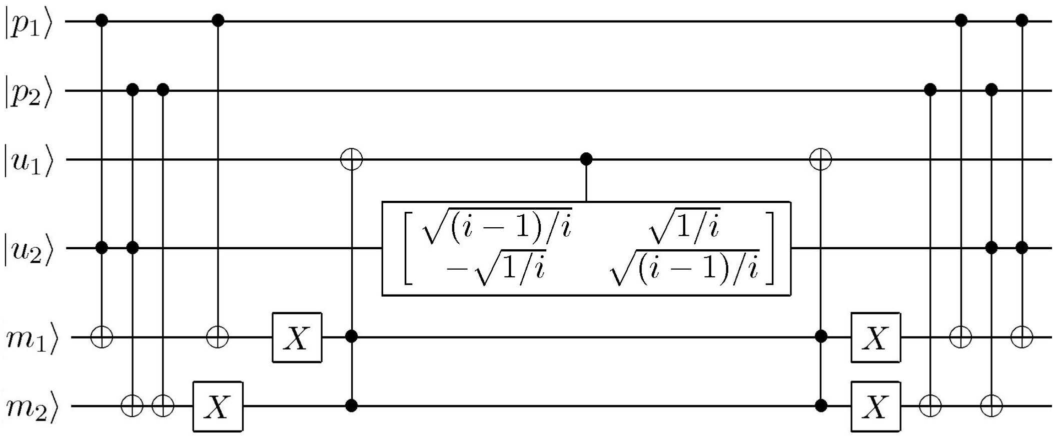
$$6.|\psi_5^i\rangle = {}^{dc}\hat{X}_{m_1 \ldots m_d u_1}\,|\psi_4^i\rangle$$

$$7.|\psi_6^i\rangle = \prod_{j=d}^{1} {}^{1c}\hat{X}_{p_j^i m_j}\,\hat{X}_{m_j}|\psi_1^i\rangle$$

$$= \frac{1}{\sqrt{p}}\sum_{k=1}^{i}|p^i; 00; p^k\rangle + \sqrt{\frac{p-i}{p}}|p^i; 01; p^i\rangle$$

$$8.|\psi_7^i\rangle = \prod_{j=d}^{1} {}^{2c}\hat{X}_{p_j^i u_2 m_j}\,|\psi_6^i\rangle$$

Circuit for storing a 2-bit pattern



[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)

QuAM storage protocol

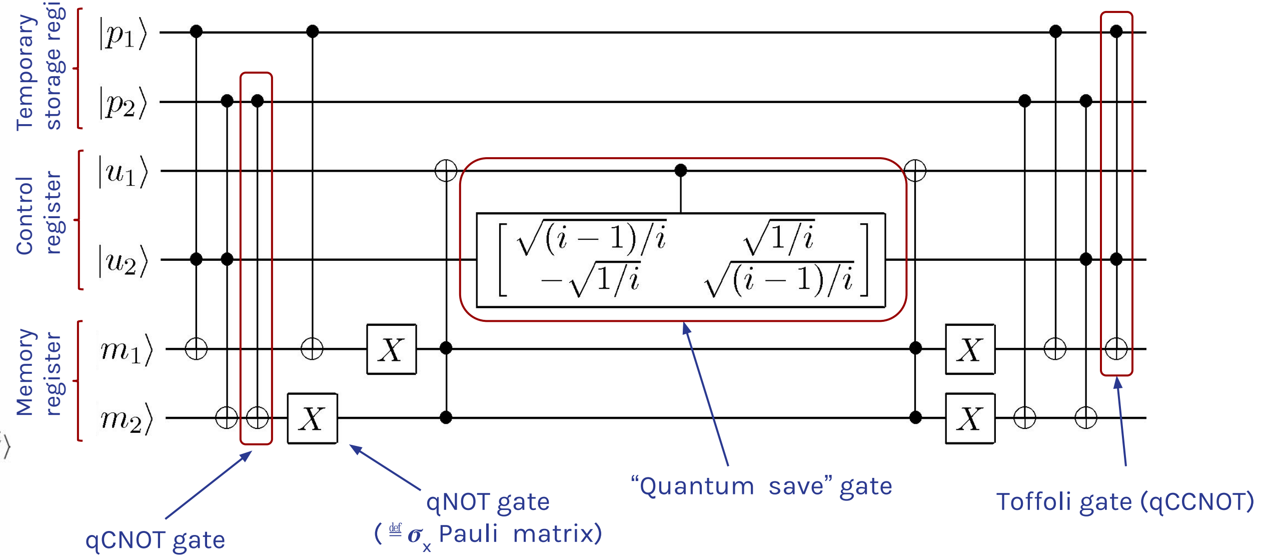
A quantum circuit implementing iterative part of the storage protocol [1].

1. $|\psi_0^1\rangle = |p_1^1, ..., p_d^1; 01; 0_1, ..., 0_d\rangle$
2. $|\psi_1^i\rangle = \prod_{j=1}^{d} {}^{2c}\hat{X}_{p_j^i u_2 m_j} |\psi_0^i\rangle$
3. $|\psi_2^i\rangle = \prod_{j=1}^{d} \hat{X}_{m_j} \, {}^{1c}\hat{X}_{p_j^i m_j} |\psi_1^i\rangle$
4. $|\psi_3^i\rangle = {}^{dc}\hat{X}_{m_1 \dots m_d u_1} |\psi_2^i\rangle$
5. $|\psi_4^i\rangle = {}^{1c}\hat{S}_{u_1 u_2}(p+1-i)|\psi_3^i\rangle$
6. $|\psi_5^i\rangle = {}^{dc}\hat{X}_{m_1 \dots m_d u_1} |\psi_4^i\rangle$
7. $|\psi_6^i\rangle = \prod_{j=d}^{1} {}^{1c}\hat{X}_{p_j^i m_j} \hat{X}_{m_j} |\psi_1^i\rangle$

$$= \frac{1}{\sqrt{p}} \sum_{k=1}^{i} |p^i; 00; p^k\rangle + \sqrt{\frac{p-i}{p}} |p^i; 01; p^i\rangle$$

8. $|\psi_7^i\rangle = \prod_{j=d}^{1} {}^{2c}\hat{X}_{p_j^i u_2 m_j} |\psi_6^i\rangle$

Circuit for storing a 2-bit pattern



[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)

QuAM storage protocol

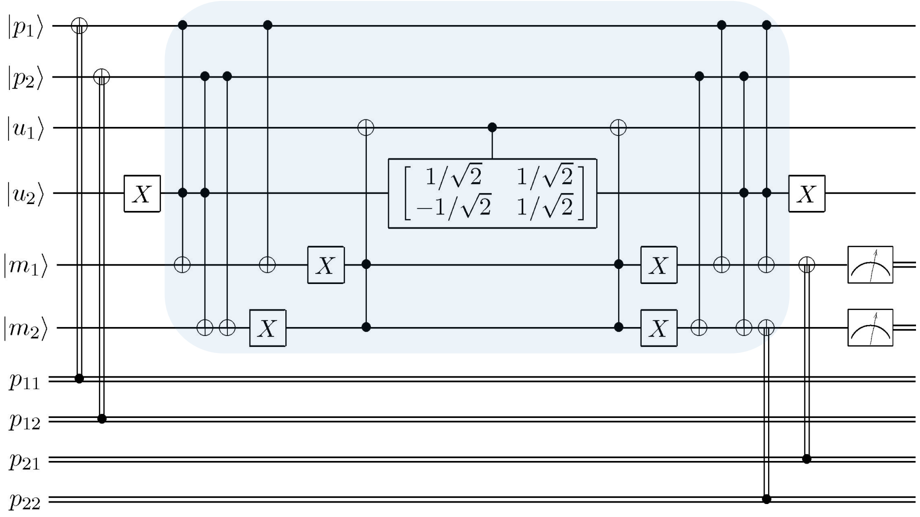
2-bit patterns example

The end-to-end circuit for storing two 2-bit patterns: “01” and “10”

“Empty” memory.

$$|\mathbf{00;00;00}\rangle$$

Classical bits for pattern read-in

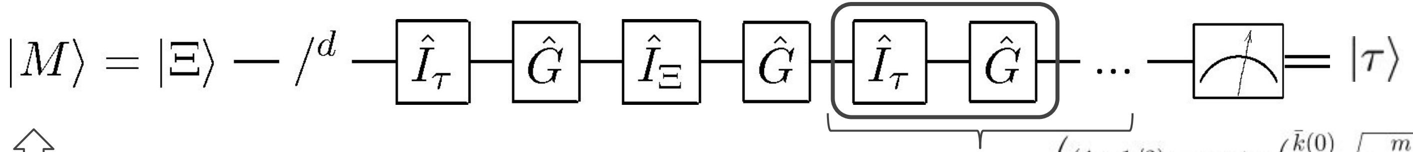


Patterns ‘01’ and ‘10’ saved.

$$\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01;00}\rangle\,\underbrace{(|\mathbf{01}\rangle + |\mathbf{10}\rangle)}$$

QuAM retrieval protocol

Generalized Grover's algorithm*



Grover's cycle. Repeated $T_j = NI\left(\frac{(j+1/2)\pi - \arctan\left(\frac{\bar{k}(0)}{\bar{l}(0)}\sqrt{\frac{m}{N-m}}\right)}{\arccos(1-\frac{2m}{N})}\right), \quad j = 0, 1, 2, \dots$ times.

$$\sum_{i=1}^{m} k_i(t)|x_i\rangle + \sum_{i=m+1}^{N} l_i(t)|x_i\rangle$$

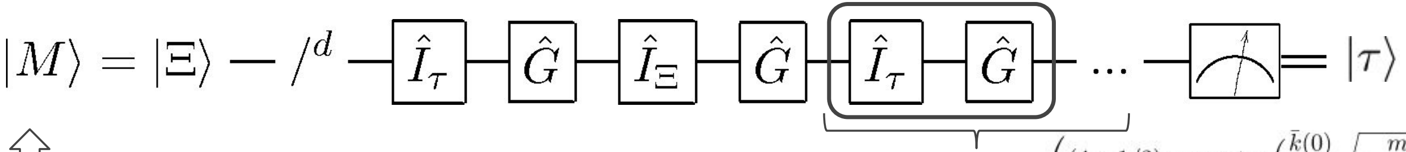
States that match the target pattern.

States that don't match the target pattern.

* $\hat{I}_\tau$ - "quantum oracle" operator. Inverts the phase of state representing the target pattern τ.
$\hat{G}$ - Grover's diffusion operator. Inverts all amplitudes about the amplitudes average.
$\hat{I}_\Xi$ - Inverts phases of all terms originally present in memory.

QuAM retrieval protocol

Generalized Grover's algorithm*



Grover's cycle. Repeated $T_j = NI\left(\frac{(j+1/2)\pi - \arctan\left(\frac{\bar{k}(0)}{\bar{l}(0)}\sqrt{\frac{m}{N-m}}\right)}{\arccos(1-\frac{2m}{N})}\right)$, $j = 0, 1, 2, \ldots$ times.

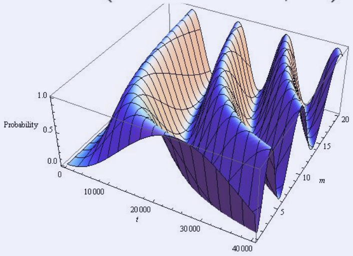
$$\sum_{i=1}^{m} k_i(t)|x_i\rangle + \sum_{i=m+1}^{N} l_i(t)|x_i\rangle$$

States that match the target pattern.

States that don't match the target pattern.

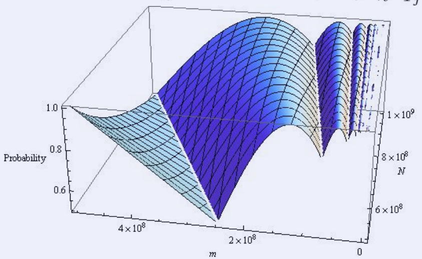
Probability "ramp-up" vs. pattern matches

$$P(t,m) = \sin^2\left((2t+1)\arcsin\sqrt{\frac{m}{N}}\right)\Bigg|_{N=10^9}$$



Peak probability vs. pattern matches and memory capacity

$$P(m,N) = \sin^2\left((2t+1)\arcsin\sqrt{\frac{m}{N}}\right)\Bigg|_{t=T_j}$$



$m = 1, N = 10^9 : T_0 = 24836, P_{max} = 0.9999999999965568$

$m = 20, N = 10^9 : T_0 = 5553, P_{max} = 0.9999999991404647$

Note: neither quantum noise, nor probabilistic memory cloning operations, are taken into account here.

* $\hat{I}_\tau$ - "quantum oracle" operator. Inverts the phase of state representing the target pattern τ.

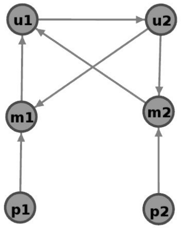
$\hat{G}$ - Grover's diffusion operator. Inverts all amplitudes about the amplitudes average.

$\hat{I}_\Xi$ - Inverts phases of all terms originally present in memory.

Topological complexity of QuAM[1]

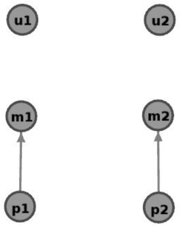
- **Storage** connectivity requirements

d=2



- **Retrieval** connectivity requirements

d=2

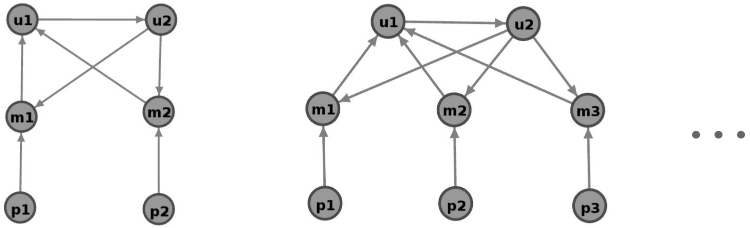


[1] **{p}**, **{u}** and **{m}** nodes represent qubits from temporary storage, control and memory registers.
d - pattern length

Topological complexity of QuAM[1]

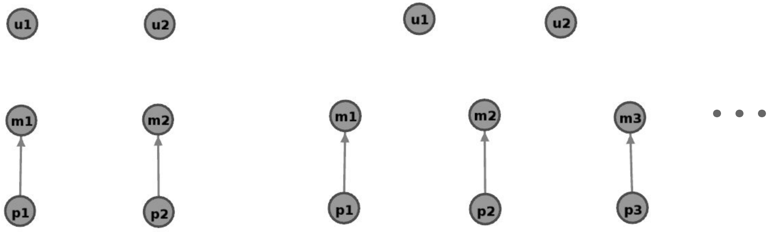
- **Storage** connectivity requirements

d=2 *d*=3



- **Retrieval** connectivity requirements

d=2 *d*=3

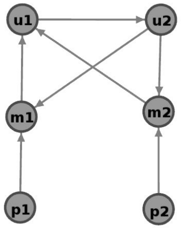


[1] **{p}**, **{u}** and **{m}** nodes represent qubits from temporary storage, control and memory registers.
d - pattern length

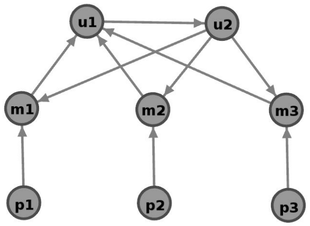
Topological complexity of QuAM[1]

- **Storage** connectivity requirements

d=2



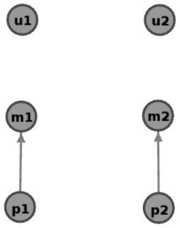
d=3



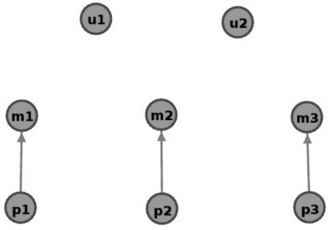
. . .

- **Retrieval** connectivity requirements

d=2



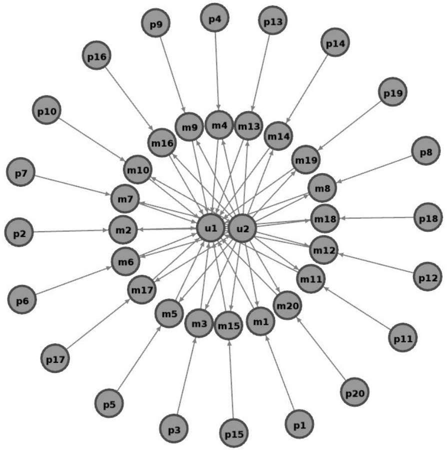
d=3



. . .

Cumulative QuAM requirements

d=20 (~ current pattern length in ATLAS)



Connectivity	*1*	*3*	*d+1*
Qubits	*d*	*d*	*2*

[1] **{p}**, **{u}** and **{m}** nodes represent qubits from temporary storage, control and memory registers.
d - pattern length

QuAM on QISKit

QISKit - Quantum Information Software Kit

An open source project comprising Python SDK, API and OpenQASM for implementing quantum algorithms on **IBM Quantum Experience (QE) hardware and simulators**.

Supported backends:

- **IBM QE cloud-based quantum chips**
  [5Q Sparrow/Raven, 16Q Albatross, 20Q]
- **Local/remote simulators**
  [with realistic noise models]

QuAM on QISKit

<u>QISKit - Quantum Information Software Kit</u>

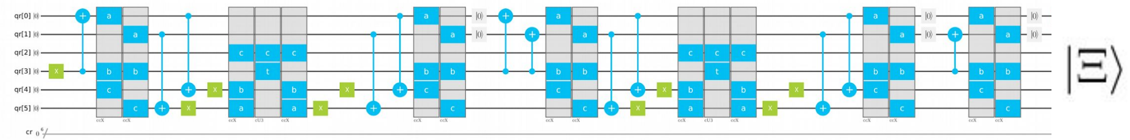
An open source project comprising Python SDK, API and OpenQASM for implementing quantum algorithms on **IBM Quantum Experience (QE) hardware and simulators**.

Supported backends:

- **IBM QE cloud-based quantum chips** [5Q Sparrow/Raven, 16Q Albatross, 20Q]
- **Local/remote simulators** [with realistic noise models]

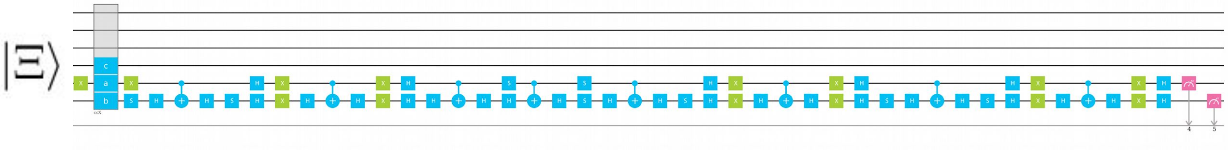
- **QuAM storage circuit generator [implemented]**

Ex.: complete circuit for encoding three 2-bit patterns



- **QuAM retrieval circuit generator [being tested]**

Ex.: complete circuit for retrieving one 2-bit pattern



Storage QASM

Snippet

```
OPENQASM 2.0;
include "qelib1.inc";
qreg qr[6];
creg cr[6];
x qr[3];
cx qr[3],qr[0];
ccx qr[0],qr[3],qr[4];
ccx qr[1],qr[3],qr[5];
cx qr[1],qr[5];
cx qr[0],qr[4];
x qr[5];
x qr[4];
ccx qr[5],qr[4],qr[2];
cu3(1.23095941734077,3.14159265358979,3.14159265358979) qr[2],qr[3]
ccx qr[5],qr[4],qr[2];
x qr[5];
x qr[4];
cx qr[1],qr[5];
cx qr[0],qr[4];
ccx qr[0],qr[3],qr[4];
ccx qr[1],qr[3],qr[5];
reset qr[0];
reset qr[1];
cx qr[3],qr[0];
cx qr[3],qr[1];
```

Retrieval QASM

Snippet

```
s qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
s qr[5];
h qr[4];
h qr[5];
x qr[4];
x qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
x qr[4];
x qr[5];
h qr[4];
h qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
```

3.

Challenges and Opportunities

Challenges and Opportunities

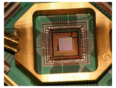
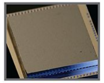
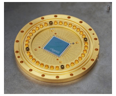
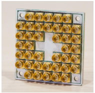
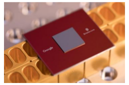
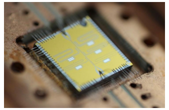
Hardware

QuAM demonstrated on

- NMR systems
- Optical systems
- D-Wave system

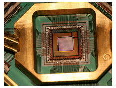
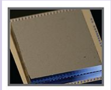
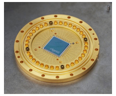
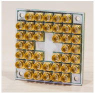
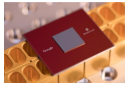
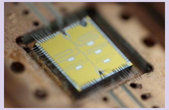
for low-order patterns. **High-order** patterns require higher qubits **connectivity** and compliant processor **topology**.

Emerging Quantum Technologies

<table>
<tr><th>Quantum Chip</th><th>Qubits</th><th>Announced</th><th>Qubit Archetype</th><th>Computing Model</th></tr>
<tr><td>D-Wave 2000Q  </td><td>2048</td><td>01/2017</td><td>Superconducting flux qubits</td><td>Quantum annealing</td></tr>
<tr><td>IBM 20Q and 50Q  </td><td>20
50</td><td>11/2017
11/2017 (tests)</td><td>Superconducting transmon qubits</td><td>Quantum circuits</td></tr>
<tr><td>Rigetti 19Q </td><td>19</td><td>12/2017</td><td>Superconducting transmon qubits</td><td>Quantum circuits</td></tr>
<tr><td>Intel Tangle Lake </td><td>49</td><td>01/2018 (tests)</td><td>Superconducting qubits [1]</td><td>Quantum circuits</td></tr>
<tr><td>⟨G|oogl|e⟩ Bristlecone </td><td>72</td><td>03/2018 (tests)</td><td>Superconducting transmon qubits</td><td>Quantum circuits</td></tr>
<tr><td>UC Berkeley QNL </td><td>4 (8)
64</td><td>2017
2022 ?</td><td>Superconducting transmon qubits</td><td>Quantum circuits</td></tr>
</table>

[1] Archetype of superconducting qubits is not disclosed. Also investing in spin qubits in silicon.

Emerging Quantum Technologies

Quantum Chip		Qubits	Announced	Qubit Archetype	Computing Model
D-Wave 2000Q	 	2048	01/2017	Superconducting **flux** qubits	Quantum annealing
IBM 20Q and 50Q	 	20	11/2017	Superconducting **transmon** qubits	Quantum circuits
		50	11/2017 (tests)		
Rigetti 19Q		19	12/2017	Superconducting **transmon** qubits	Quantum circuits
Intel Tangle Lake		49	01/2018 (tests)	Superconducting qubits [1]	Quantum circuits
⟨G|oogl|e⟩ Bristlecone		72	03/2018 (tests)	Superconducting **transmon** qubits	Quantum circuits
UC Berkeley QNL		4 (8)	2017	Superconducting **transmon** qubits	Quantum circuits
		64	2022 ?		

[1] Archetype of superconducting qubits is not disclosed. Also investing in spin qubits in silicon.

Challenges and Opportunities

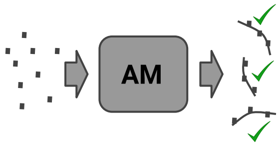
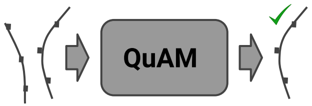
Hardware

QuAM demonstrated on

- NMR systems
- Optical systems
- D-Wave system

for low-order patterns. **High-order** patterns require higher qubits **connectivity** and compliant processor **topology**.

Functional trade-offs

AM **generates**, **completes**, and **validates** track patterns:	QuAM **completes** and **validates** track patterns:
	

Challenges and Opportunities

Hardware

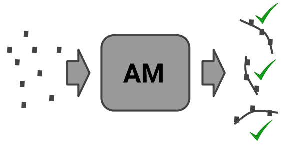
QuAM demonstrated on

- NMR systems
- Optical systems
- D-Wave system

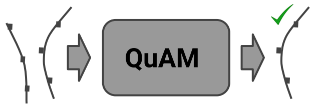
for low-order patterns. **High-order** patterns require higher qubits **connectivity** and compliant processor **topology**.

Functional trade-offs

AM **generates, completes,** and **validates** track patterns:



QuAM **completes** and **validates** track patterns:



Memory persistence

Memory state collapses with each query. Repetitive re-initialization is a show stopper. A possible solution may employ **probabilistic cloning of memory** reducing efficiency.

Summary

- **QC paradigm** can yield **asymmetrical advantages** in handling certain challenges of HL/HE HEP real-time track pattern recognition
- **QuAM** features:
  - **Exponential storage capacity**
  - ***Optimal*** QA for pattern **recall**
- Current status:
  - **Theoretical analysis** of QuAM properties **completed**
    - Memory initialization iterations
    - Recall probability bounds
    - Topological complexity analysis
  - **Storage/retrieval** quantum circuit generators **implemented** in QISKit
    - Ready to run on real quantum hardware
- Coming soon:
  - QuAM on the latest quantum hardware (targeting IBM QE chips)
  - QuAM performance tests (timing, efficiency)