Hacking 2 048 RSA code with 8 100 qubits and a multimode memory with 2 hours storage time

Élie Gouzien and Nicolas Sangouard

December 3, 2020, GDR-IQFA'11 Colloquium

Institut de Physique Théorique, CEA Saclay

Quantum computing hardware

Possible harware for quantum computing
Superconducting qubits, trapped atoms/ions, optics, spins...

Quantum computing hardware

Possible harware for quantum computing
Superconducting qubits, trapped atoms/ions, optics, spins...

Superconducting qubits

- ✓ high gate fidelity
- ✓ scalable

Quantum computing hardware

Possible harware for quantum computing
Superconducting qubits, trapped atoms/ions, optics, spins...

Superconducting qubits

- ✓ high gate fidelity
- ✓ scalable
- ✗ short coherence time
- ✗ 2D connectivity

Quantum computing hardware

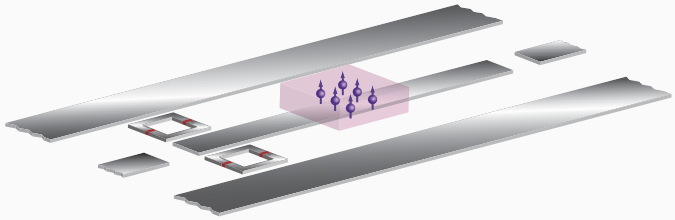
Possible harware for quantum computing

Superconducting qubits, trapped atoms/ions, optics, spins...

Superconducting qubits

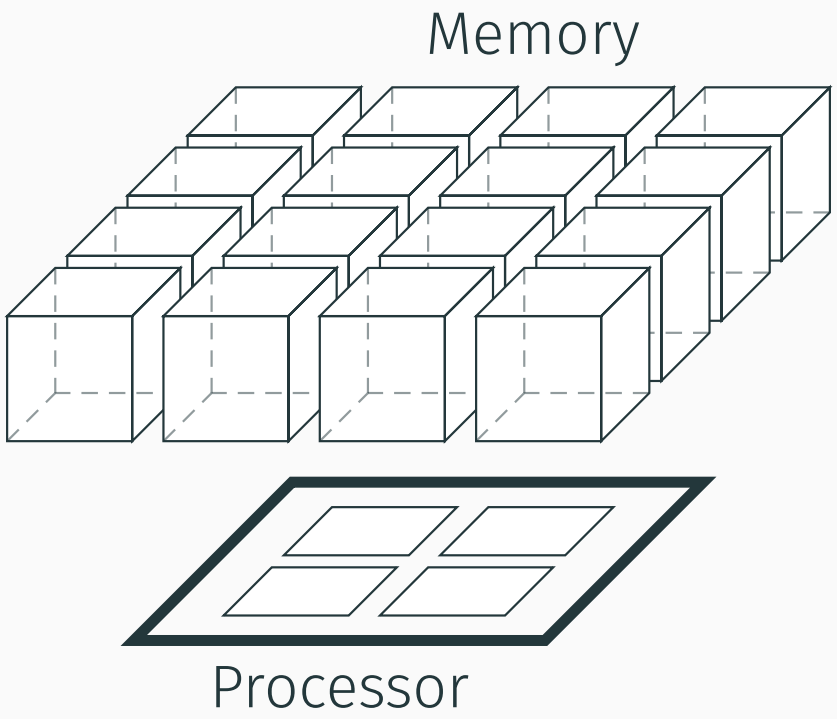
- ✓ high gate fidelity
- ✓ scalable
- ✗ short coherence time
- ✗ 2D connectivity

Superconducting processor + memory

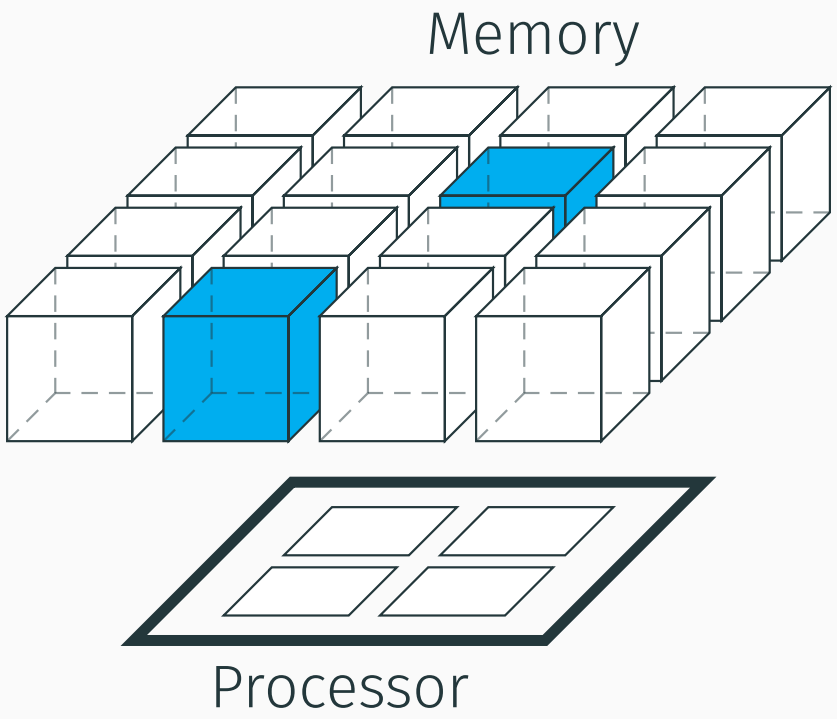


Cécile Grezes et al. “Towards a spin-ensemble quantum memory for superconducting qubits”. *Comptes Rendus Physique* 17.7 (2016), pp. 693–704

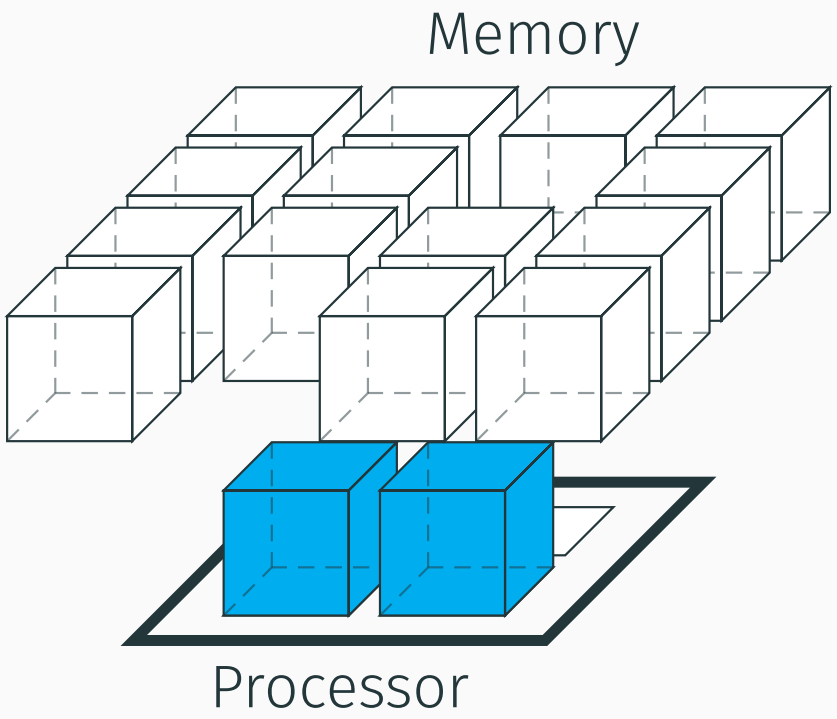
Proposed architecture



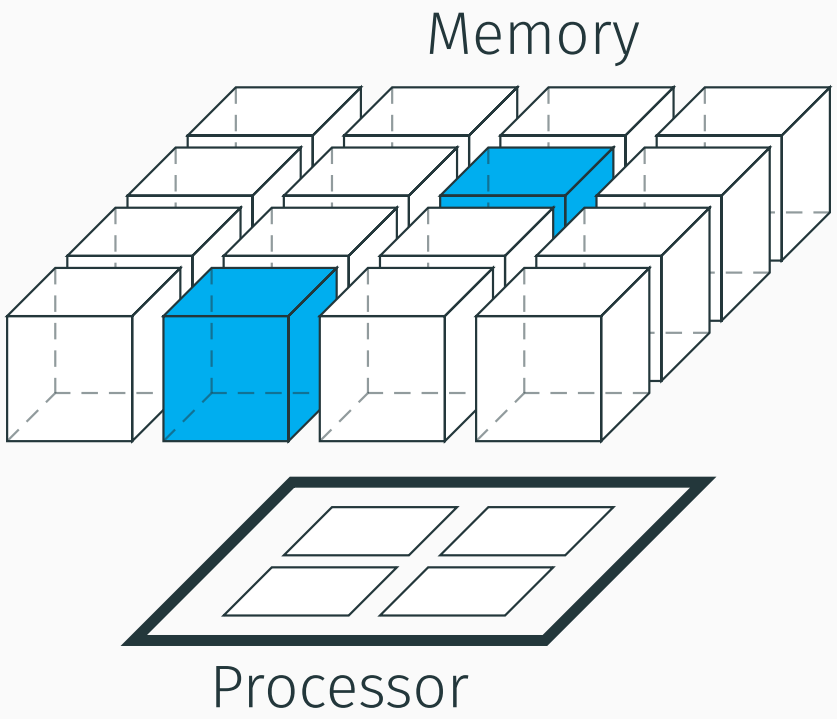
Proposed architecture



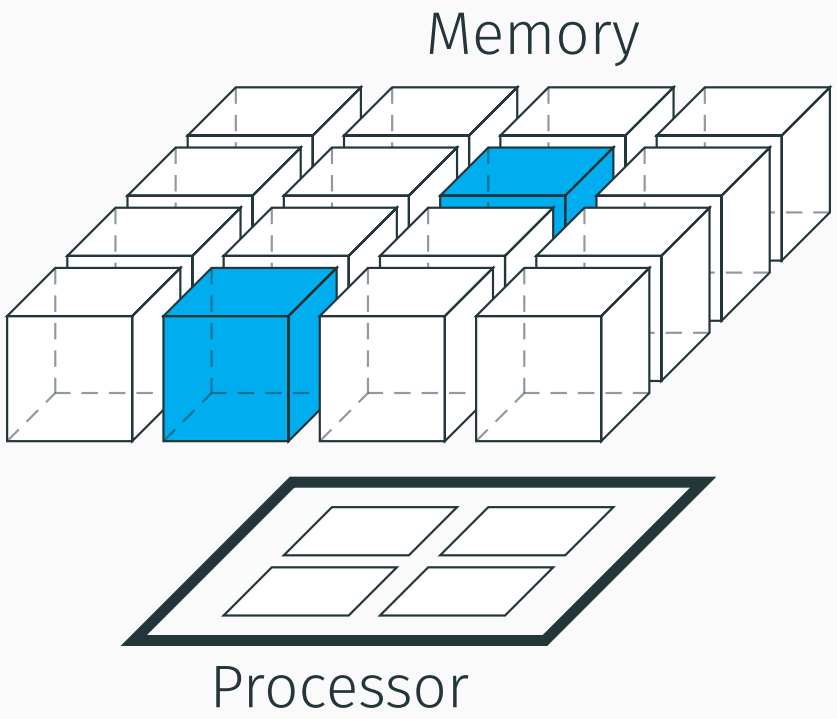
Proposed architecture



Proposed architecture



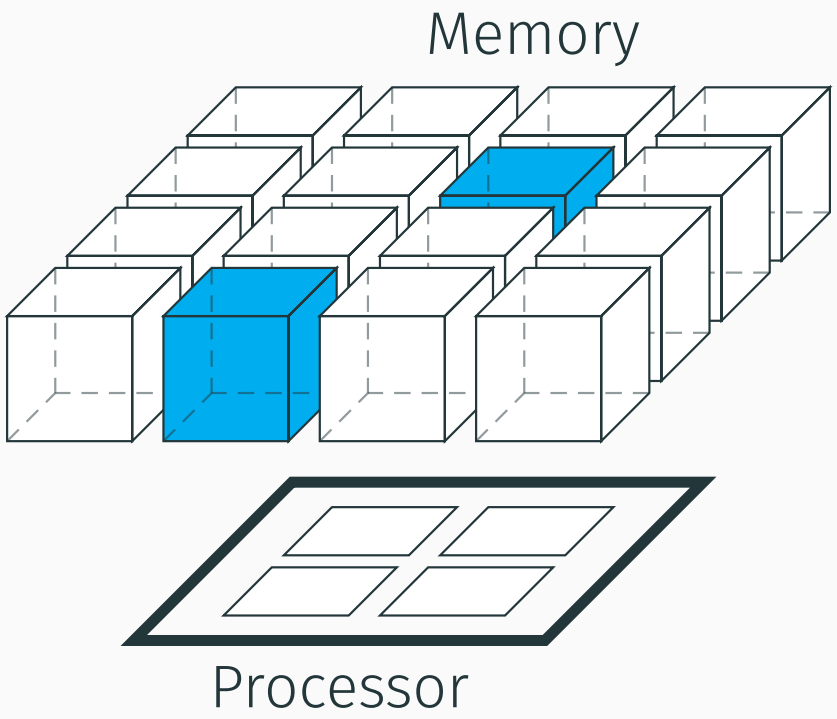
Proposed architecture



Potential benefits

- small processor
- better connectivity
- reduced decoherence for in-memory qubits

Proposed architecture



Potential benefits

- small processor
- better connectivity
- reduced decoherence for in-memory qubits

What is quantitatively the advantage brought by this architecture?

RSA number prime factorization

Shor's algorithm (and variants)

- classically hard: no polynomial algorithm
- breaking RSA shows importance of quantum communications
- check the correct operation of the machine
- mostly sequential

RSA number prime factorization

Shor's algorithm (and variants)

- classically hard: no polynomial algorithm
- breaking RSA shows importance of quantum communications
- check the correct operation of the machine
- mostly sequential

Factorization with 2D superconducting processor

Craig Gidney and Martin Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". 2019

Factorization algorithm

Shor's algorithm

Problem
$N = pq$, p and q prime numbers

Shor's algorithm

Problem
$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly

Shor's algorithm

Problem

$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly
- find the period r of $e \mapsto g^e \mod N$

Shor's algorithm

Problem

$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly
- find the period r of $e \mapsto g^e \mod N$
- hypothesis r even and $g^{\frac{r}{2}} \not\equiv N - 1 \mod N$

Shor's algorithm

Problem

$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly
- find the period r of $e \mapsto g^e \mod N$
- hypothesis r even and $g^{\frac{r}{2}} \not\equiv N - 1 \mod N$
- $g^r \equiv 1 \mod N \Leftrightarrow \left(g^{\frac{r}{2}} - 1\right)\left(g^{\frac{r}{2}} + 1\right) \equiv 0 \mod N$

Shor's algorithm

Problem

$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly
- find the period r of $e \mapsto g^e \mod N$
- hypothesis r even and $g^{\frac{r}{2}} \not\equiv N - 1 \mod N$
- $g^r \equiv 1 \mod N \Leftrightarrow \left(g^{\frac{r}{2}} - 1\right)\left(g^{\frac{r}{2}} + 1\right) \equiv 0 \mod N$
- $p, q = \gcd[g^{\frac{r}{2}} - 1, N], \gcd[g^{\frac{r}{2}} + 1, N]$

Shor's algorithm

Problem

$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly
- find the period r of $e \mapsto g^e \mod N$
- hypothesis r even and $g^{\frac{r}{2}} \not\equiv N - 1 \mod N$
- $g^r \equiv 1 \mod N \Leftrightarrow \left(g^{\frac{r}{2}} - 1\right)\left(g^{\frac{r}{2}} + 1\right) \equiv 0 \mod N$
- $p, q = \gcd[g^{\frac{r}{2}} - 1, N], \gcd[g^{\frac{r}{2}} + 1, N]$

Run on quantum hardware

1. prepare $\frac{1}{\sqrt{2^{n_e}}} \sum_{x=0}^{2^{n_e}-1} |x\rangle$, n_e number of bits of the exponent
2. compute $e \mapsto g^e \mod N$
3. quantum Fourier transform to recover r

Shor's algorithm

Problem

$N = pq$, p and q prime numbers

Principle

- pick up $g \in \mathbb{Z}_N^*$ randomly
- find the period r of $e \mapsto g^e \mod N$
- hypothesis r even and $g^{\frac{r}{2}} \not\equiv N - 1 \mod N$
- $g^r \equiv 1 \mod N \Leftrightarrow \left(g^{\frac{r}{2}} - 1\right)\left(g^{\frac{r}{2}} + 1\right) \equiv 0 \mod N$
- $p, q = \gcd[g^{\frac{r}{2}} - 1, N], \gcd[g^{\frac{r}{2}} + 1, N]$

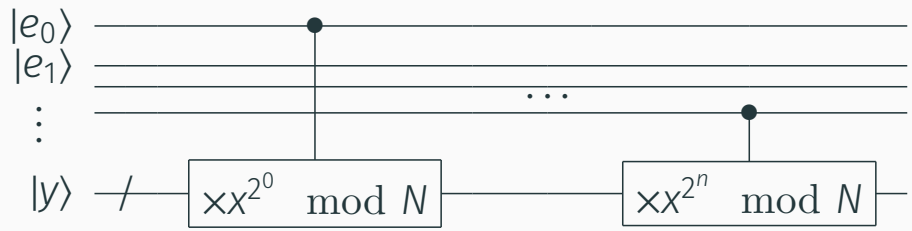
Run on quantum hardware

1. prepare $\frac{1}{\sqrt{2^{n_e}}} \sum_{x=0}^{2^{n_e}-1} |x\rangle$, n_e number of bits of the exponent
2. compute $e \mapsto g^e \mod N$
3. quantum Fourier transform to recover r

Standard implementation complexity

Exponentiation: $|e\rangle|y\rangle \mapsto |e\rangle\left|y \times x^e \mod N\right\rangle$

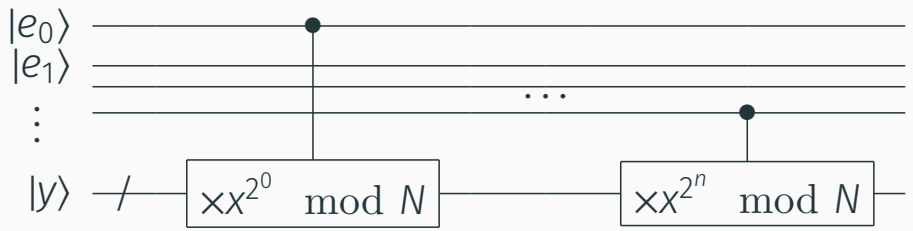
$$x^e = x^{\sum_i 2^i e_i} = \prod_i \left[x^{2^i}\right]^{e_i}$$



Standard implementation complexity

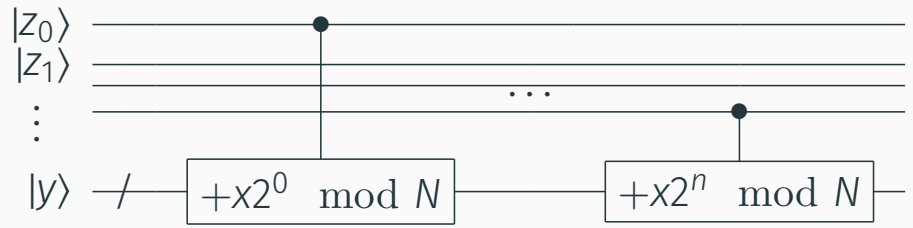
Exponentiation: $|e\rangle|y\rangle \mapsto |e\rangle\left|y \times x^e \mod N\right\rangle$

$$x^e = x^{\sum_i 2^i e_i} = \prod_i \left[x^{2^i}\right]^{e_i}$$



Multiplication: $|z\rangle|y\rangle \mapsto |z\rangle|y + x \times z \mod N\rangle$

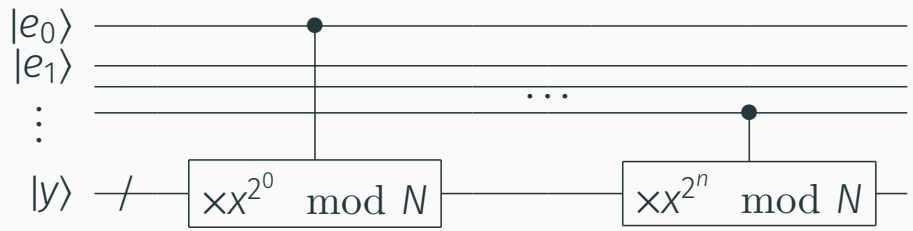
$$xz = x\sum_i 2^i z_i = \sum_i \left(x2^i\right) z_i$$



Standard implementation complexity

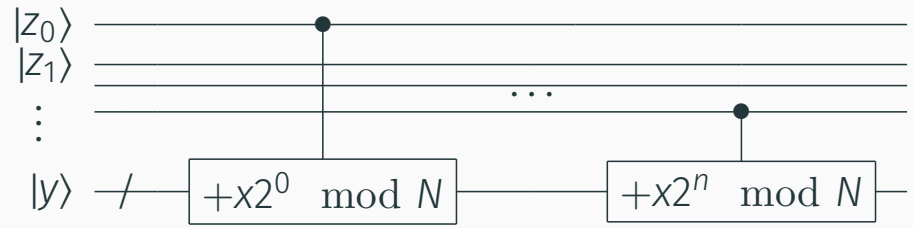
Exponentiation: $|e\rangle|y\rangle \mapsto |e\rangle|y \times x^e \mod N\rangle$

$$x^e = x^{\sum_i 2^i e_i} = \prod_i \left[x^{2^i}\right]^{e_i}$$



Multiplication: $|z\rangle|y\rangle \mapsto |z\rangle|y + x \times z \mod N\rangle$

$$xz = x\sum_i 2^i z_i = \sum_i \left(x2^i\right) z_i$$

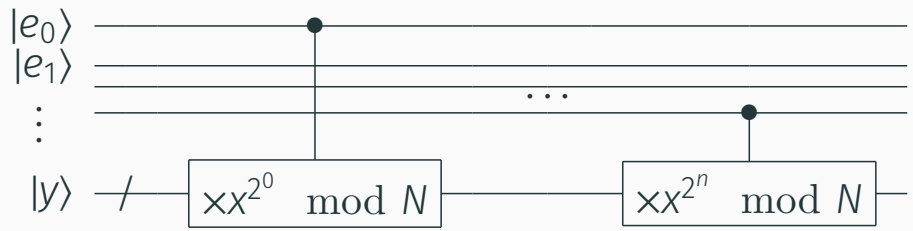


Addition: $|x\rangle|y\rangle \mapsto |x\rangle|x + y \mod N\rangle$
$O(n)$ with carry-ripple addition
Steven A. Cuccaro et al. “A new quantum ripple-carry addition circuit”. 2004

Standard implementation complexity

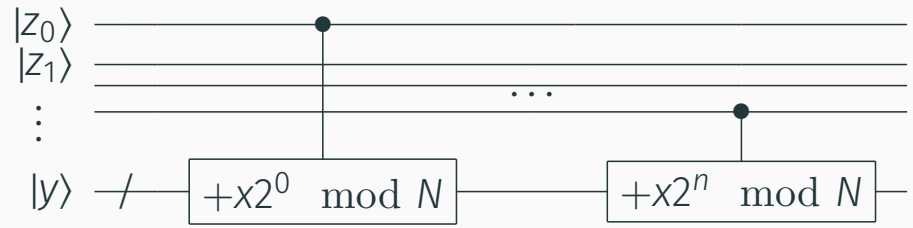
Exponentiation: $|e\rangle|y\rangle \mapsto |e\rangle|y \times x^e \mod N\rangle$

$$x^e = x^{\sum_i 2^i e_i} = \prod_i \left[x^{2^i}\right]^{e_i}$$



Multiplication: $|z\rangle|y\rangle \mapsto |z\rangle|y + x \times z \mod N\rangle$

$$xz = x\sum_i 2^i z_i = \sum_i \left(x2^i\right) z_i$$



Addition: $|x\rangle|y\rangle \mapsto |x\rangle|x + y \mod N\rangle$
$O(n)$ with carry-ripple addition

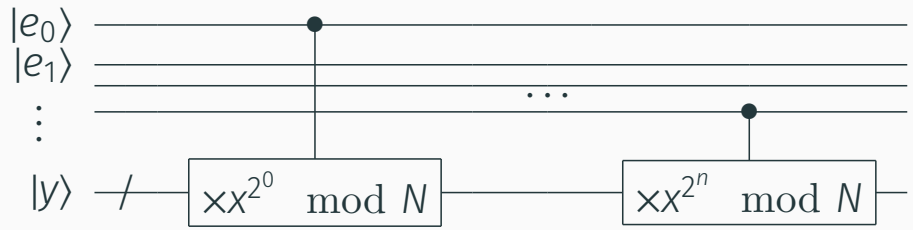
Steven A. Cuccaro et al. “A new quantum ripple-carry addition circuit”. 2004

Modular addition: addition, comparison, controlled correction and clean-up of ancillary qubits: $O(n)$

Standard implementation complexity

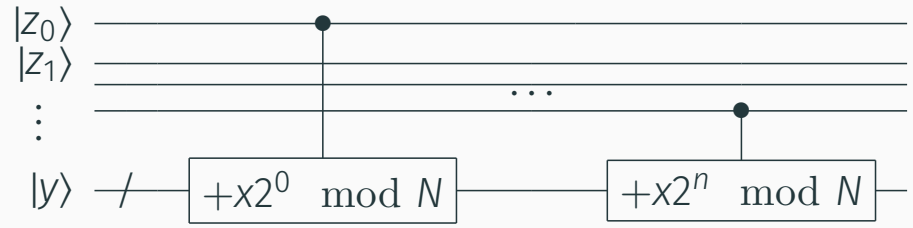
Exponentiation: $|e\rangle|y\rangle \mapsto |e\rangle|y \times x^e \mod N\rangle$

$$x^e = x^{\sum_i 2^i e_i} = \prod_i \left[x^{2^i}\right]^{e_i}$$



Multiplication: $|z\rangle|y\rangle \mapsto |z\rangle|y + x \times z \mod N\rangle$

$$xz = x\sum_i 2^i z_i = \sum_i \left(x2^i\right) z_i$$



Addition: $|x\rangle|y\rangle \mapsto |x\rangle|x + y \mod N\rangle$
$O(n)$ with carry-ripple addition

Steven A. Cuccaro et al. "A new quantum ripple-carry addition circuit". 2004

Modular addition: addition, comparison, controlled correction and clean-up of ancillary qubits: $O(n)$

Complexity

Modular exponentiation: $O\left(n^3\right)$

Advanced algorithm complexity

Ekerå and Håstad's algorithm

Size of the exponent: $1.5n$ instead of $2n$ for Shor's algorithm

Advanced algorithm complexity

Ekerå and Håstad's algorithm

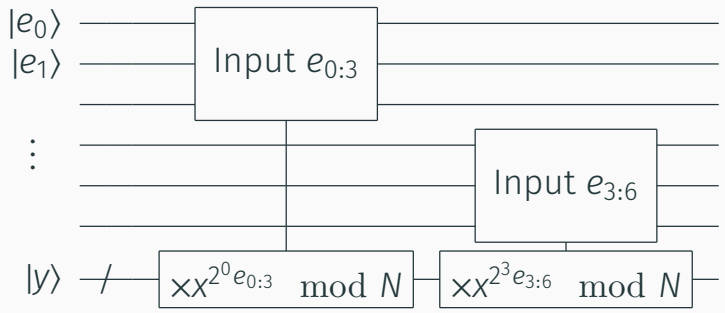
Size of the exponent: $1.5n$ instead of $2n$ for Shor's algorithm

Windowed arithmetic circuits

Craig Gidney. "Windowed quantum arithmetic". 2019

Exponentiation example:

$$e = \sum_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} 2^i e_{i:i+w}, \qquad x^e = \prod_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} x^{2^i e_{i:i+w}}$$



Advanced algorithm complexity

Ekerå and Håstad's algorithm

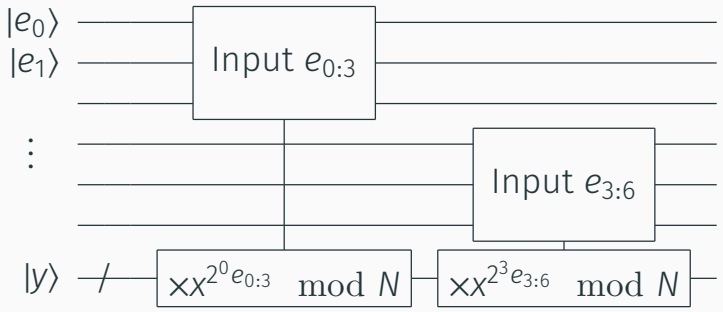
Size of the exponent: $1.5n$ instead of $2n$ for Shor's algorithm

Windowed arithmetic circuits

Craig Gidney. "Windowed quantum arithmetic". 2019

Exponentiation example:

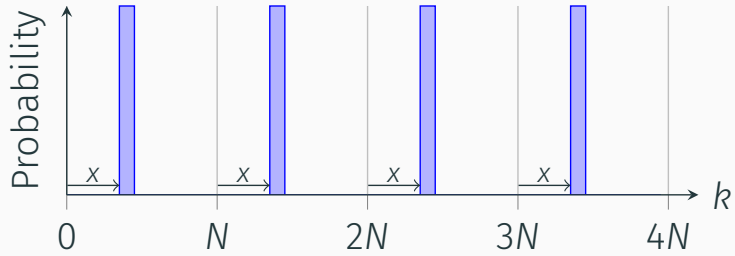
$$e = \sum_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} 2^i e_{i:i+w}, \qquad x^e = \prod_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} x^{2^i e_{i:i+w}}$$



Modular addition through coset representation

Craig Gidney. "Approximate encoded permutations and piecewise quantum adders". 2019

x represented by $\frac{1}{\sqrt{2^c}} \sum_{i=0}^{2^c-1} |x + iN\rangle$



Advanced algorithm complexity

Ekerå and Håstad's algorithm

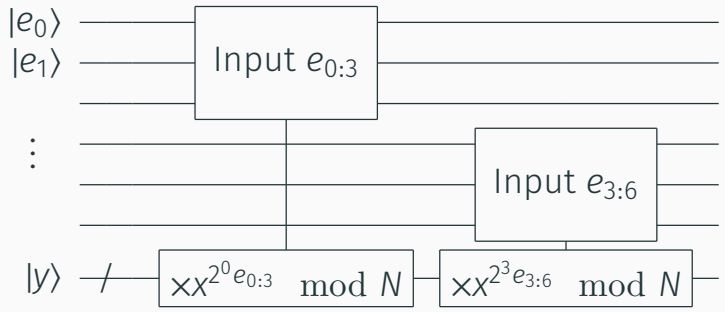
Size of the exponent: $1.5n$ instead of $2n$ for Shor's algorithm

Windowed arithmetic circuits

Craig Gidney. "Windowed quantum arithmetic". 2019

Exponentiation example:

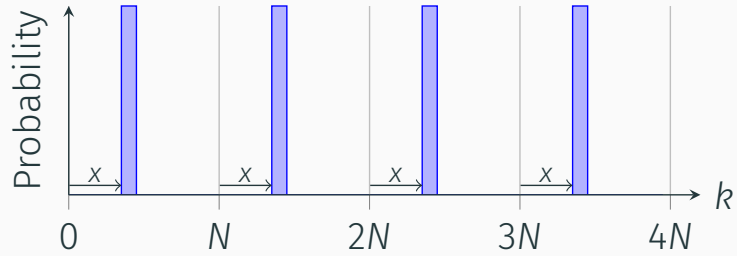
$$e = \sum_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} 2^i e_{i:i+w}, \quad x^e = \prod_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} x^{2^i e_{i:i+w}}$$



Modular addition through coset representation

Craig Gidney. "Approximate encoded permutations and piecewise quantum adders". 2019

x represented by $\frac{1}{\sqrt{2^c}} \sum_{i=0}^{2^c-1} |x + iN\rangle$



Complexity

parallel depth: $O\left(\frac{n^2\left(n+2^{2w}\right)}{w^2}\right)$, gates: $O\left(\frac{n^3 2^{2w}}{w^2}\right)$

Advanced algorithm complexity

Ekerå and Håstad's algorithm

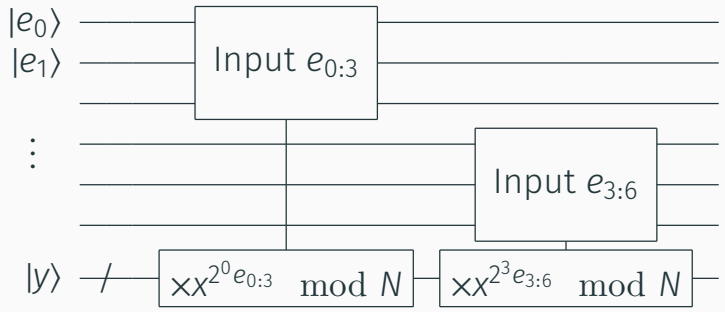
Size of the exponent: $1.5n$ instead of $2n$ for Shor's algorithm

Windowed arithmetic circuits

Craig Gidney. "Windowed quantum arithmetic". 2019

Exponentiation example:

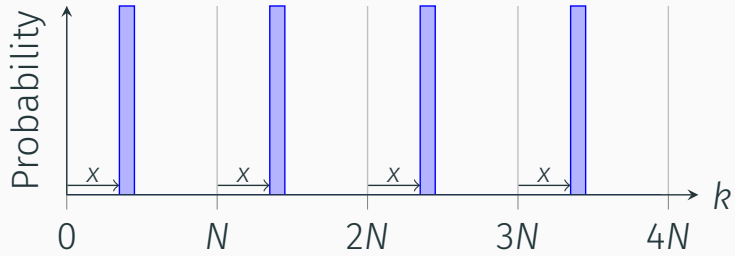
$$e = \sum_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} 2^i e_{i:i+w}, \quad x^e = \prod_{\substack{0 \le i < n \\ i \equiv 0 \mod w}} x^{2^i e_{i:i+w}}$$



Modular addition through coset representation

Craig Gidney. "Approximate encoded permutations and piecewise quantum adders". 2019

x represented by $\frac{1}{\sqrt{2^c}} \sum_{i=0}^{2^c-1} |x + iN\rangle$



Complexity

parallel depth: $O\left(\frac{n^2\left(n+2^{2w}\right)}{w^2}\right)$, gates: $O\left(\frac{n^3 2^{2w}}{w^2}\right)$; choice: $w = O(1)$

$\Rightarrow O\left(n^3\right)$ but better pre-factors than basic algorithm

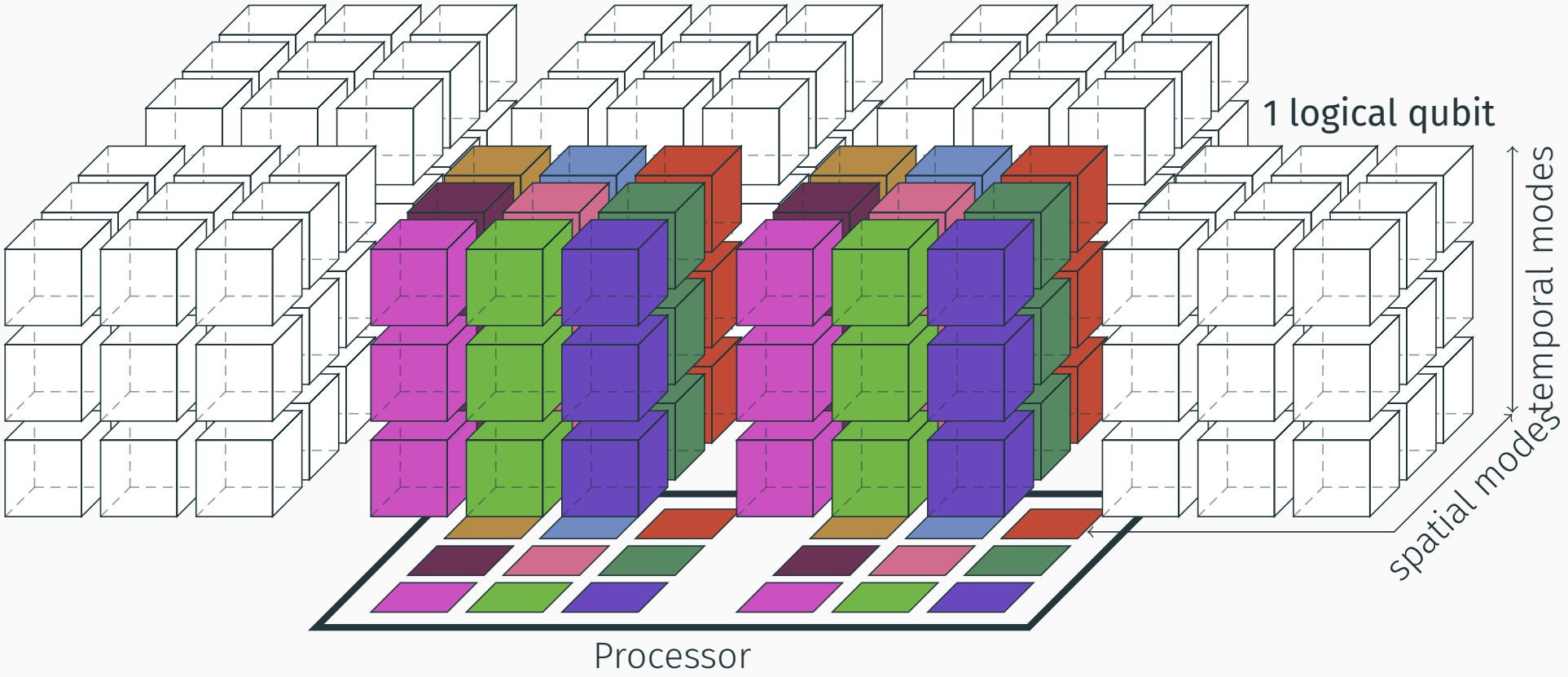
Error correction

3D gauge color codes

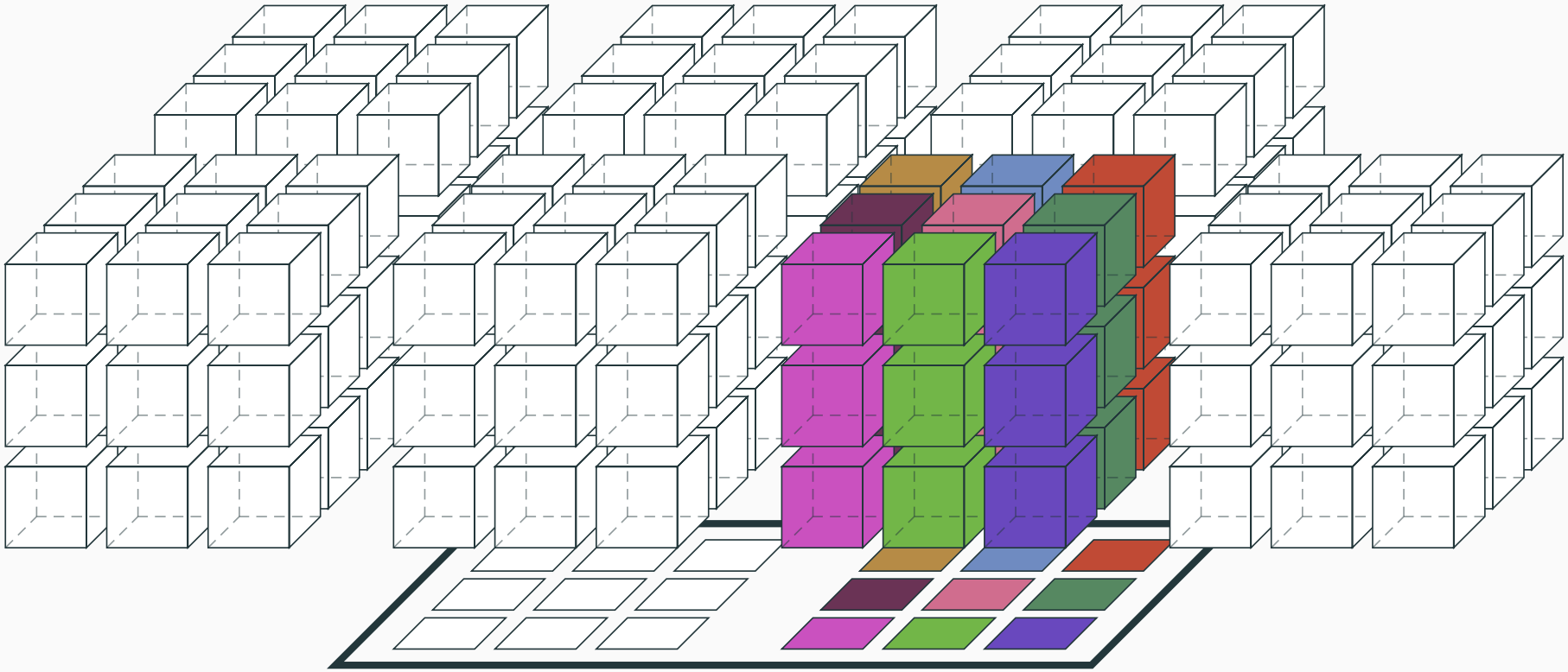
- H, CNOT and T gates transversal
  $\Rightarrow$ universal set
  $\Rightarrow$ no distillation/gate teleportation
- 3D code, 2D processor
  $\Longrightarrow$ 2D parallelism for stabilizers measurements
- single-shot error correction: no need to repeat stabilizers measurement
- Error threshold: 0.75 %

Héctor Bombín. "Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes". *New Journal of Physics* 17.8 (2015), p. 083002

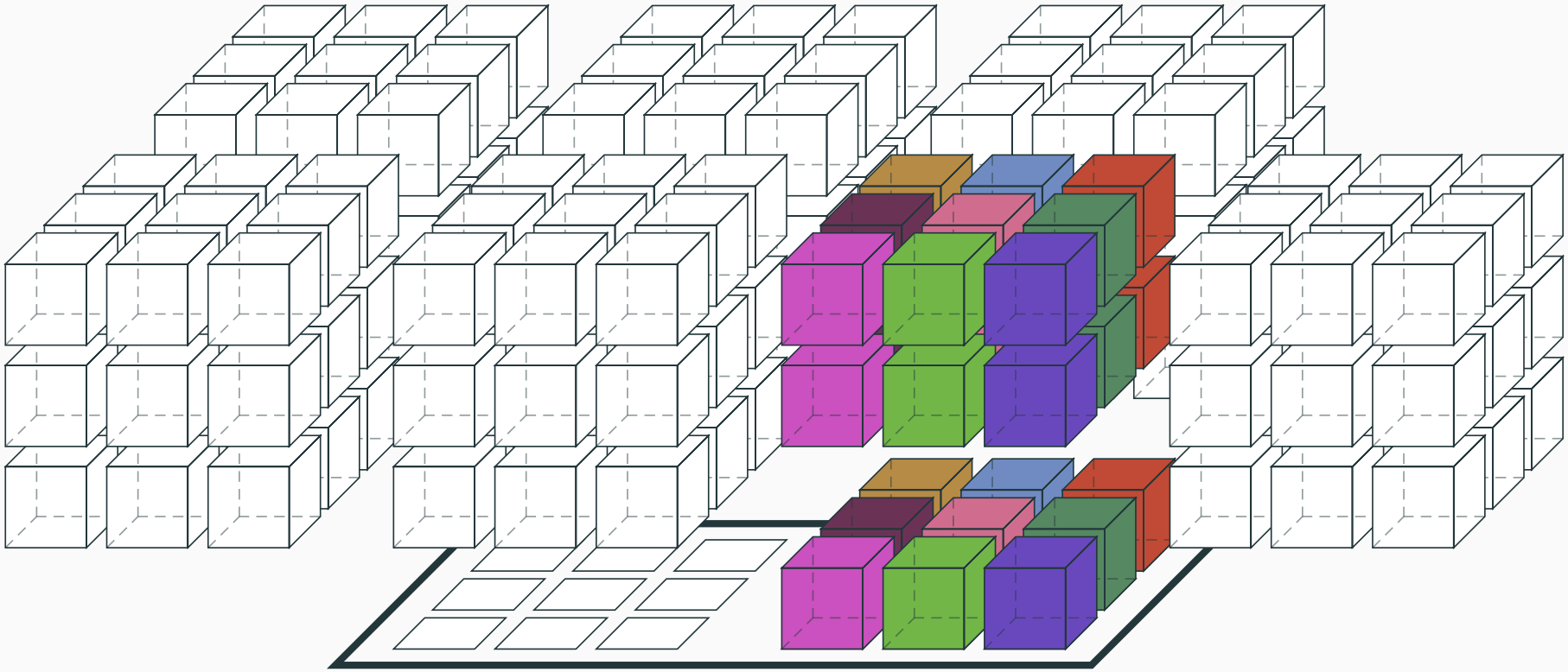
How to measure syndromes



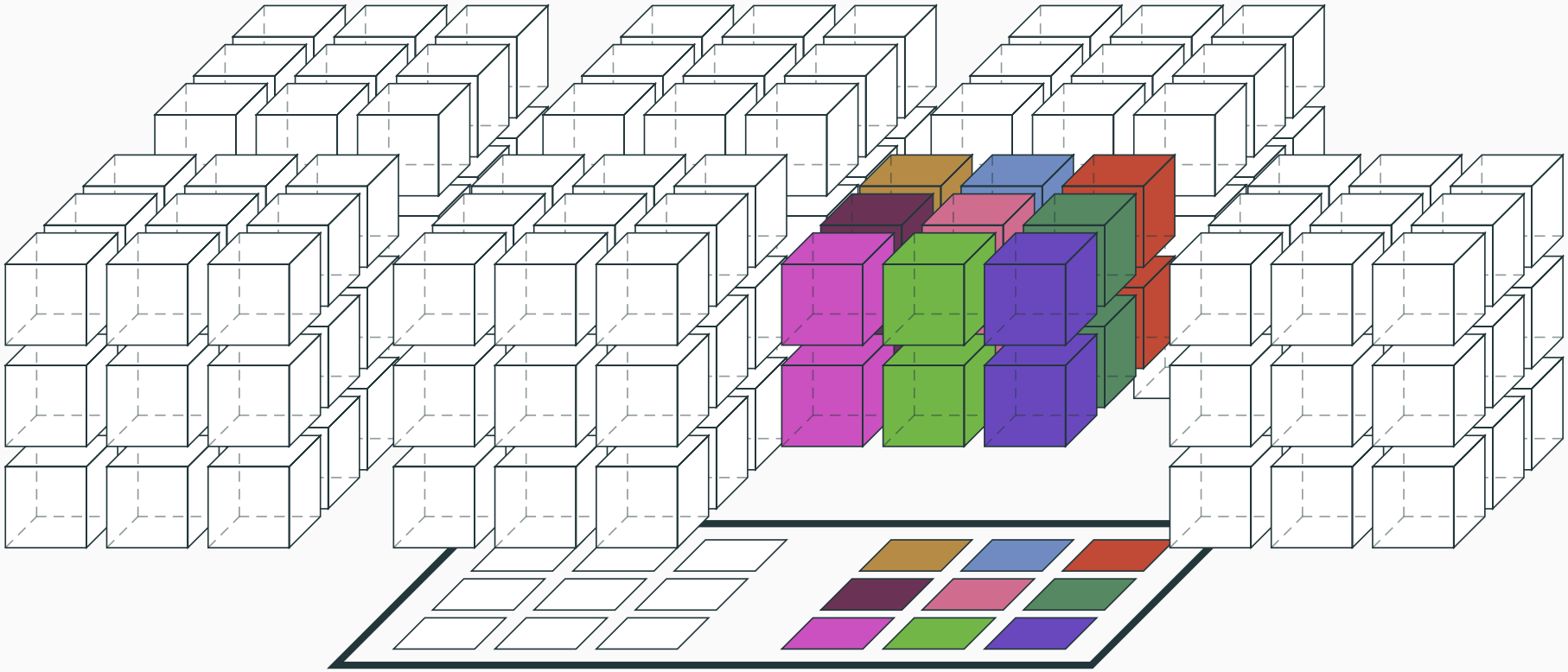
How to measure syndromes

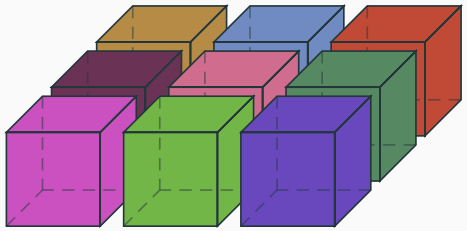


How to measure syndromes

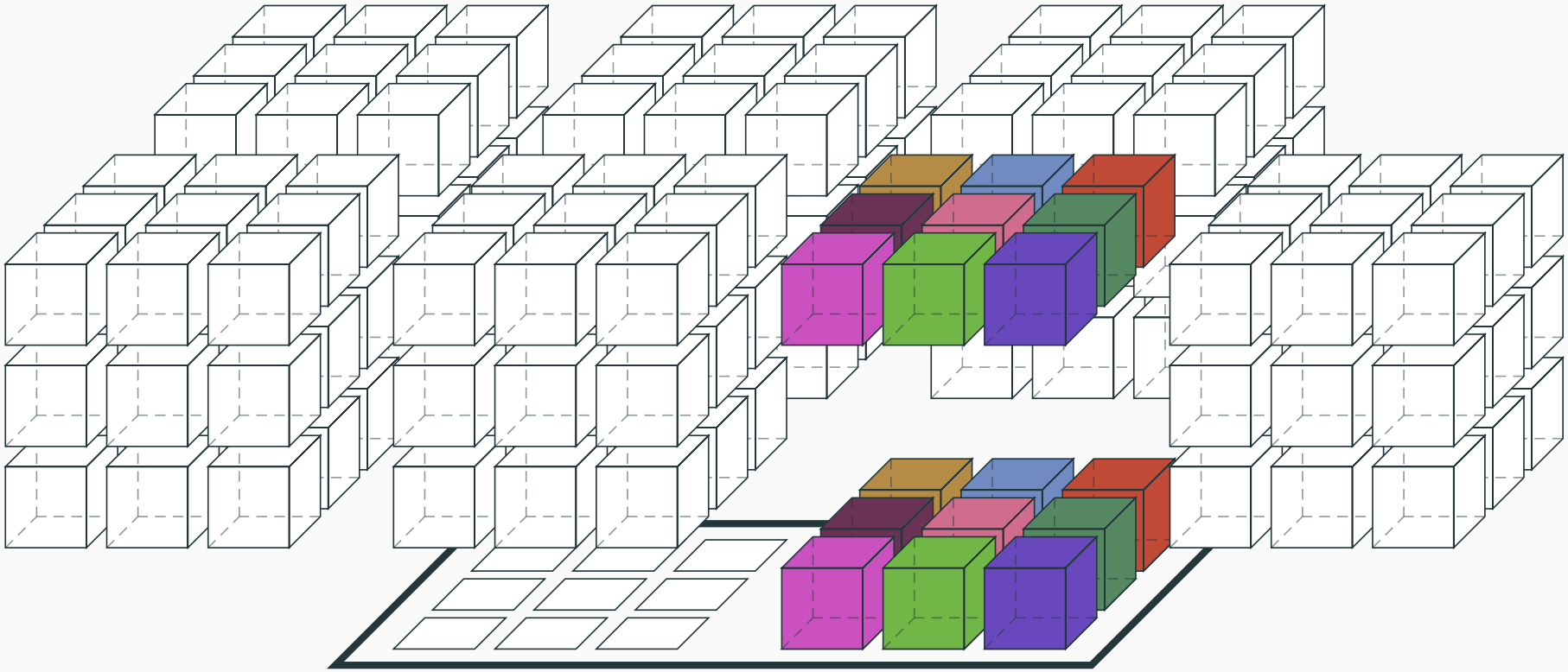


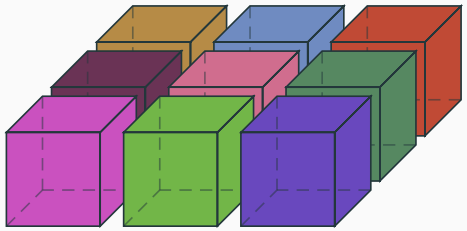
How to measure syndromes



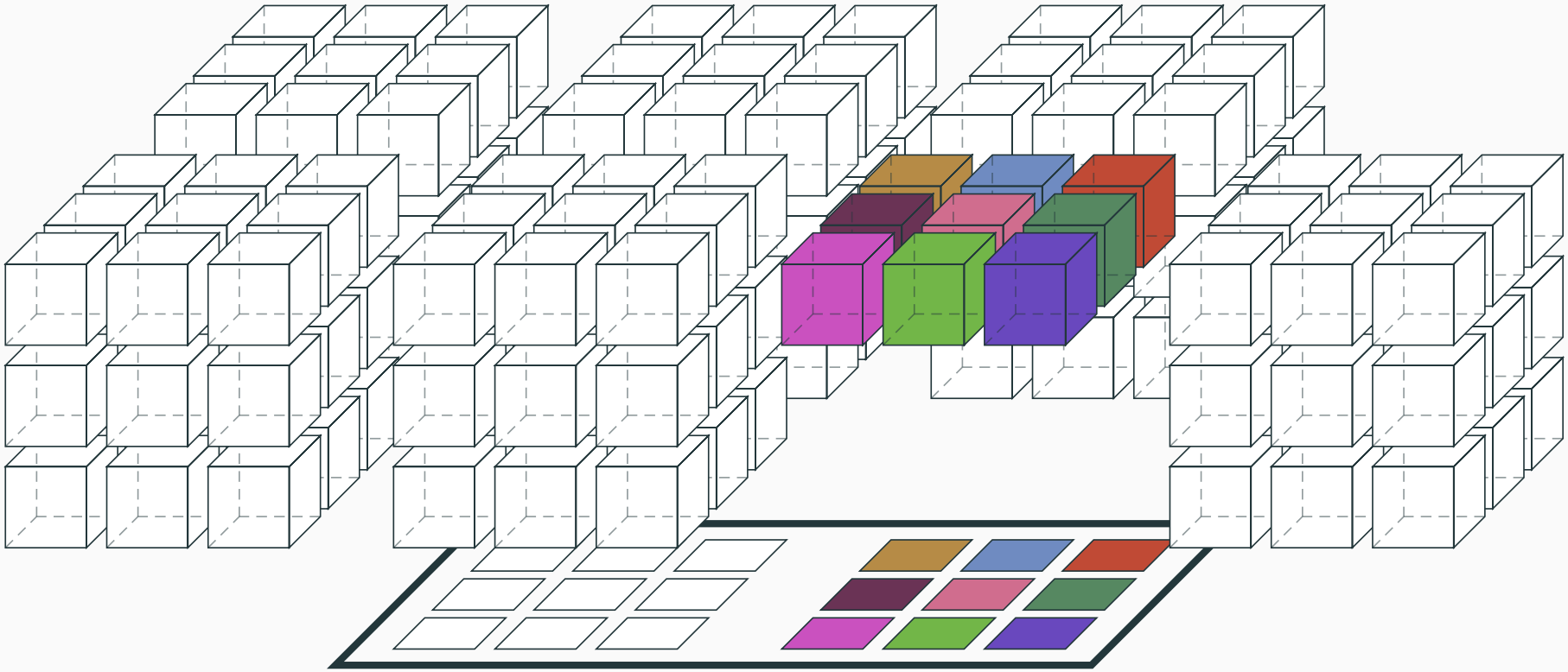


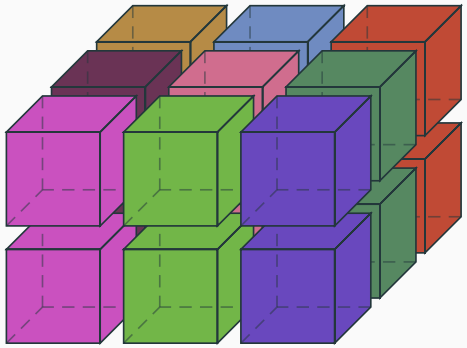
How to measure syndromes



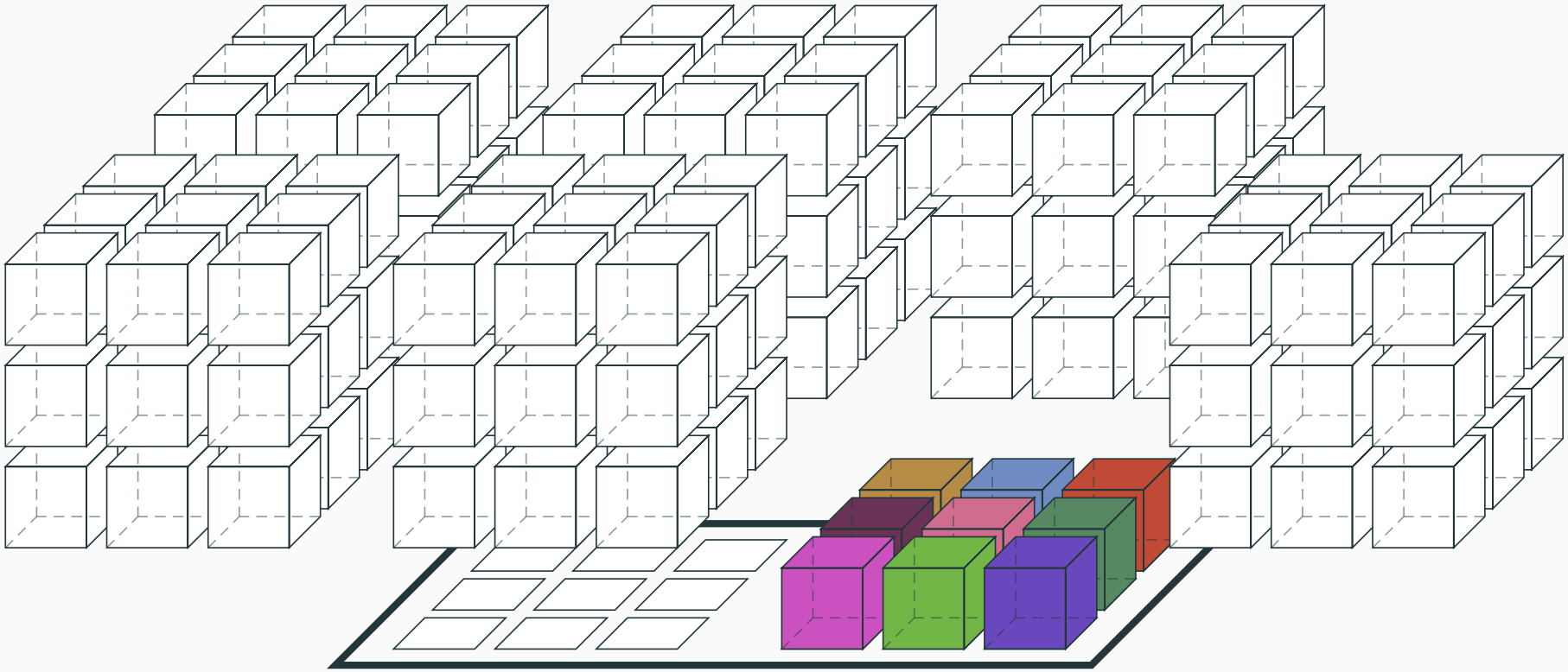


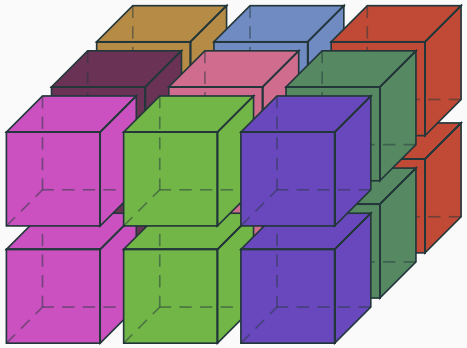
How to measure syndromes



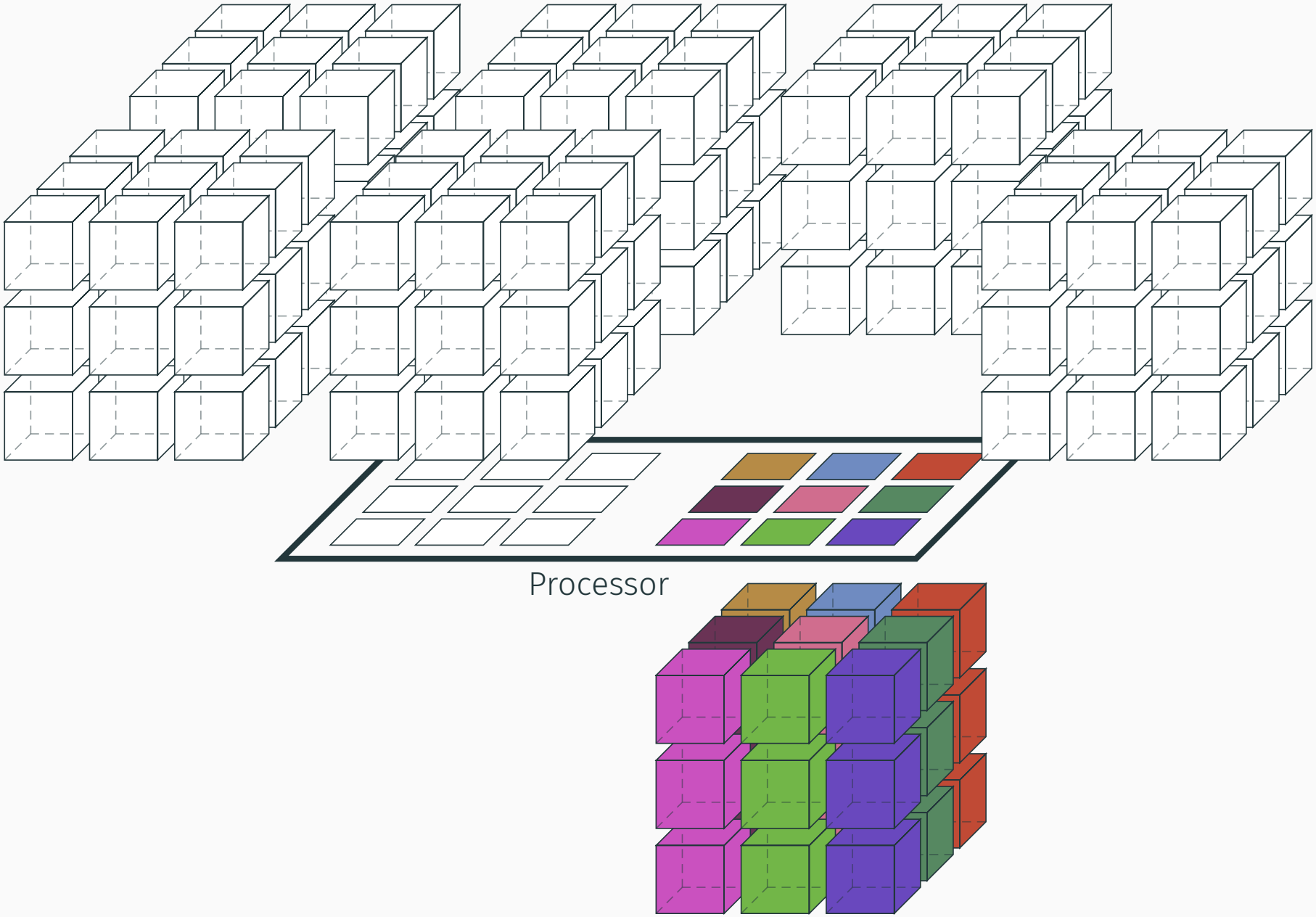


How to measure syndromes



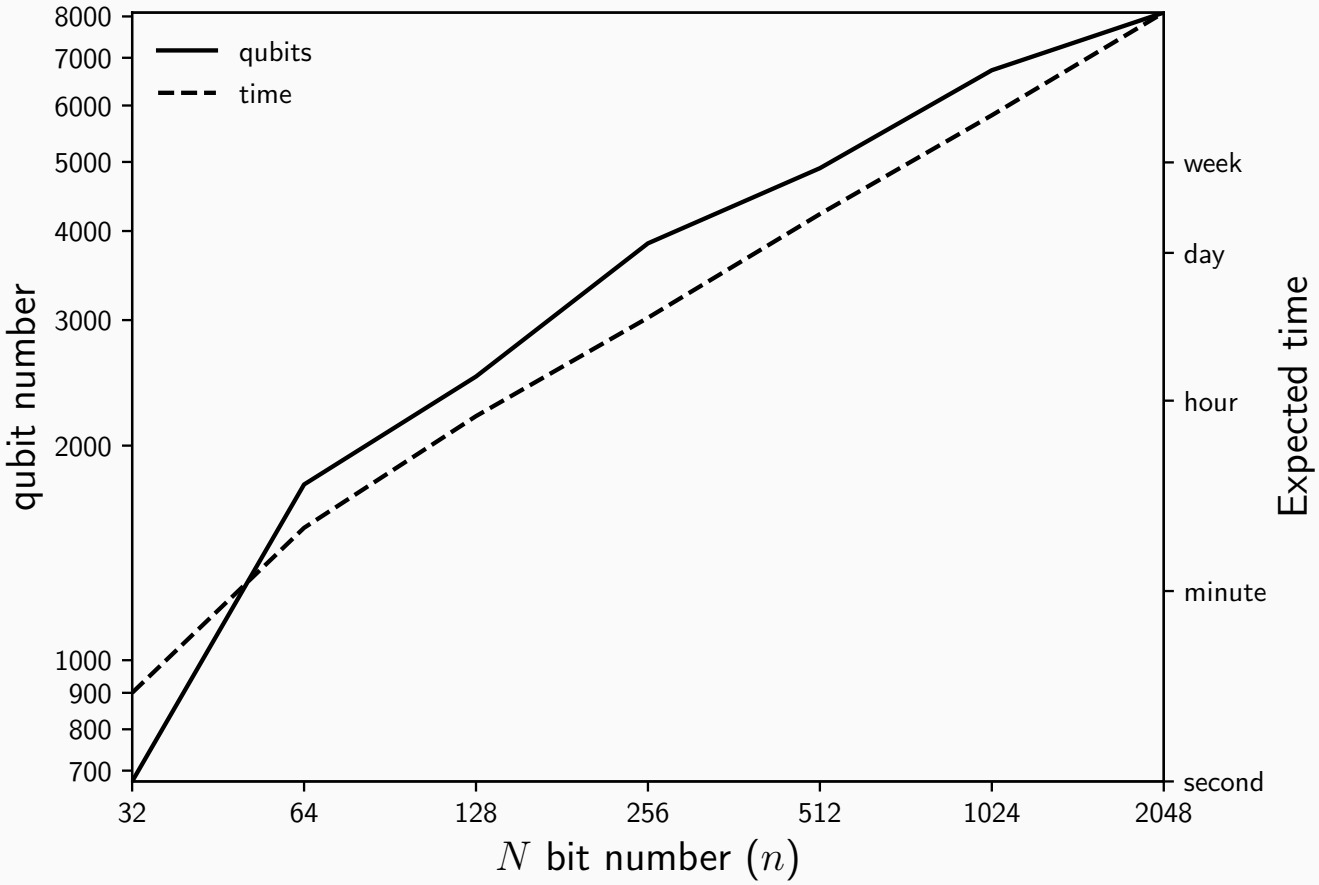


How to measure syndromes

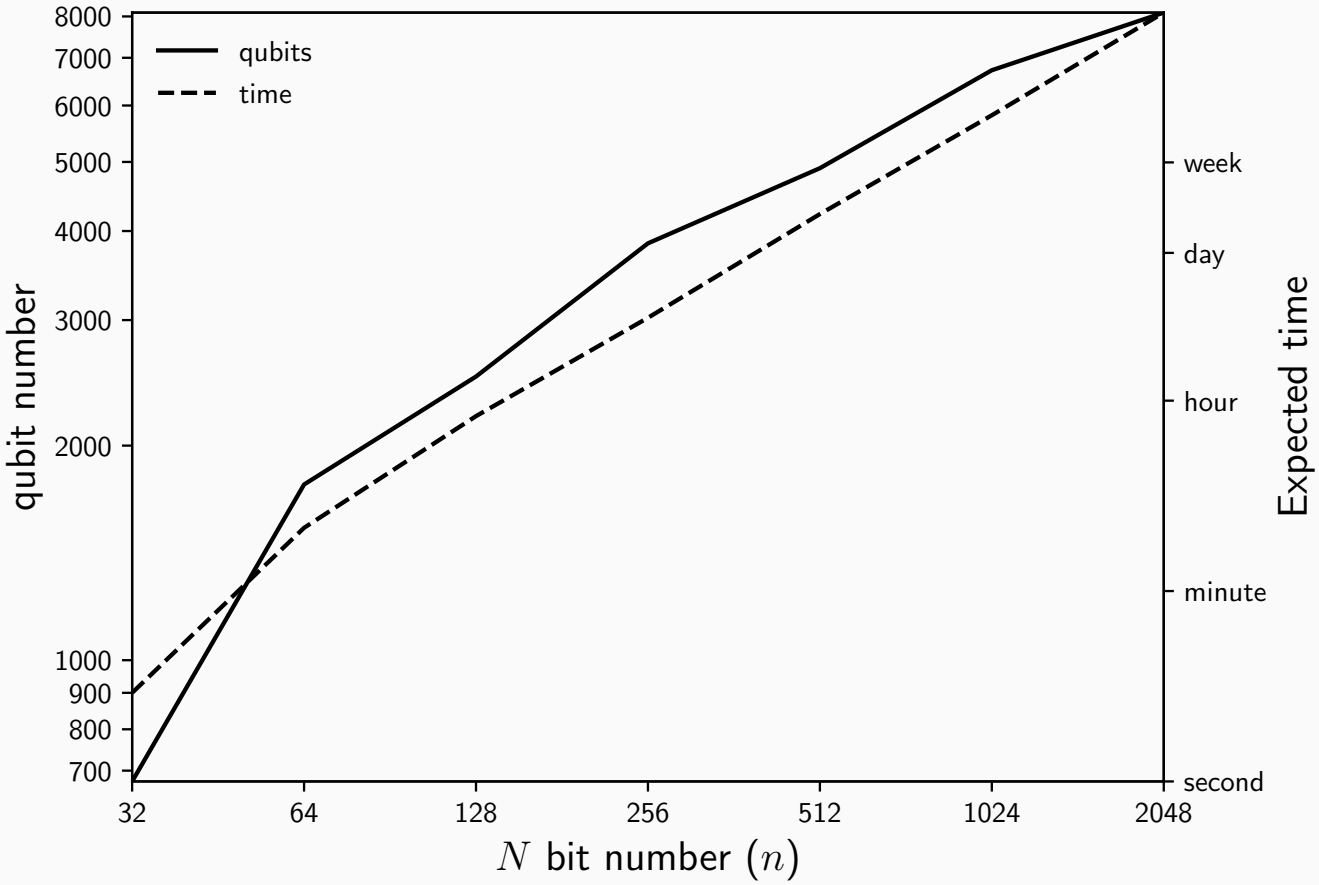


Results

Resources required for factorization

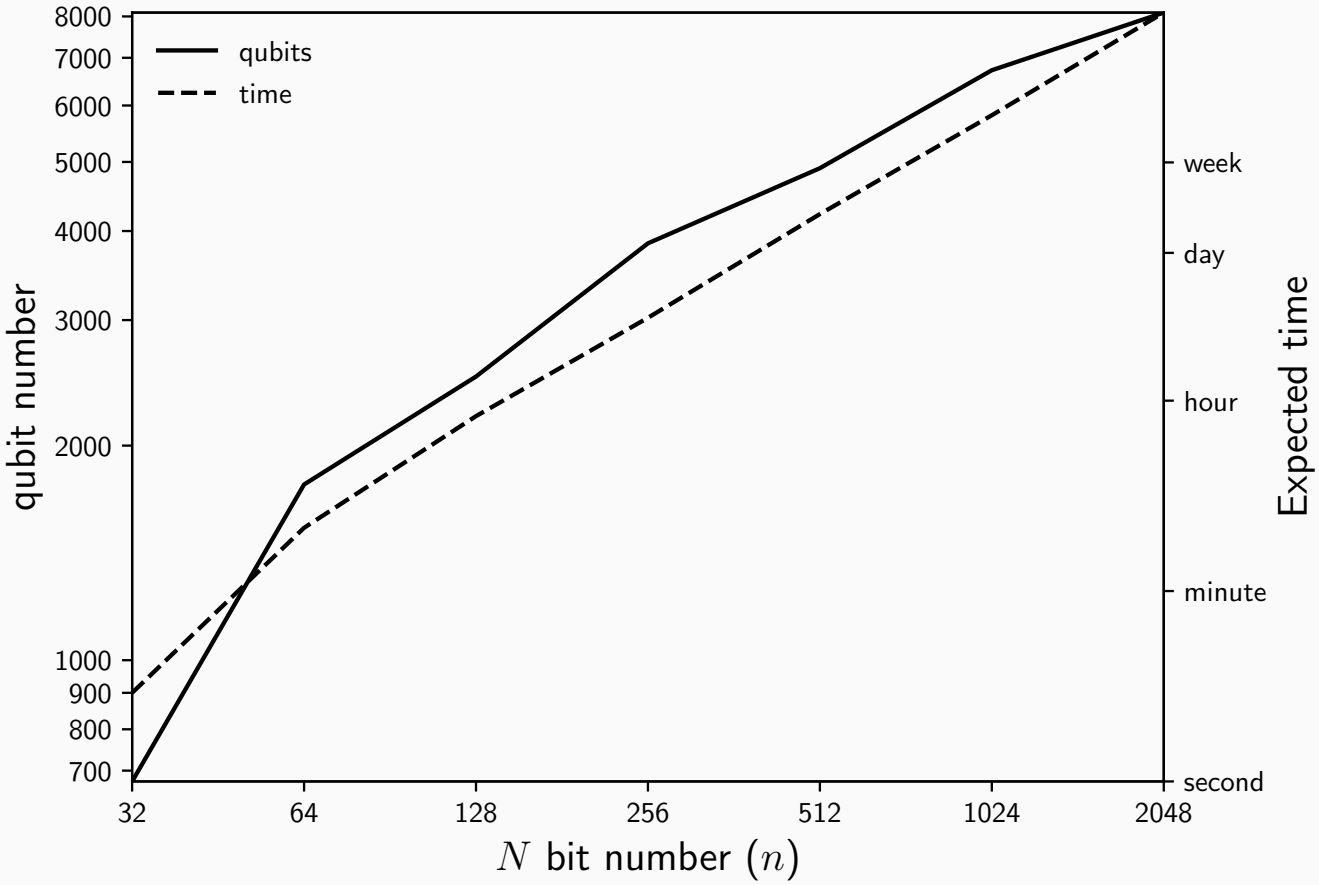


Resources required for factorization



RSA 829 bits factorization
6 084 qubits, 10 days; maximum storage time: $\approx$ 15 min

Resources required for factorization



RSA 829 bits factorization
6 084 qubits, 10 days; maximum storage time: ≈ 15 min

RSA 2 048 bits factorization
8 100 qubits, 175 days; maximum storage time: ≈ 2 h

Comparison with 2D processor without memory

Only two logical qubit slices in the processor
9 277 logical qubits for RSA-2 048, only slices of 2 logical qubits in the superconducting processor.

Comparison with 2D processor without memory

Only two logical qubit slices in the processor
9 277 logical qubits for RSA-2 048, only slices of 2 logical qubits in the superconducting processor.

Better than 2D connectivity between physical qubits
Error-corrected universal set of gates

Comparison with 2D processor without memory

Only two logical qubit slices in the processor
9 277 logical qubits for RSA-2 048, only slices of 2 logical qubits in the superconducting processor.

Better than 2D connectivity between physical qubits
Error-corrected universal set of gates

Arbitrary connectivity between logical qubits
2D processor with well-chosen layout is already efficient

Comparison with 2D processor without memory

Only two logical qubit slices in the processor
9 277 logical qubits for RSA-2 048, only slices of 2 logical qubits in the superconducting processor.

Better than 2D connectivity between physical qubits
Error-corrected universal set of gates

Arbitrary connectivity between logical qubits
2D processor with well-chosen layout is already efficient

Reduced decoherence in the memory
Not really important because error-correction is exponentially efficient

Implementation and perspectives

Implementation and perspectives

Implementation

- spin echo memory
- multimode cavity
- microwave photon router

Implementation and perspectives

Implementation

- spin echo memory
- multimode cavity
- microwave photon router

Additionnal improvement

- logical circuit more adapted to sequential computation
- more efficient error correction

Implementation and perspectives

Implementation

- spin echo memory
- multimode cavity
- microwave photon router

Additionnal improvement

- logical circuit more adapted to sequential computation
- more efficient error correction

Other problems

- optimization
- chemistry

The end

Thanks for your attention