http://www.sfc.wide.ad.jp/aqua/

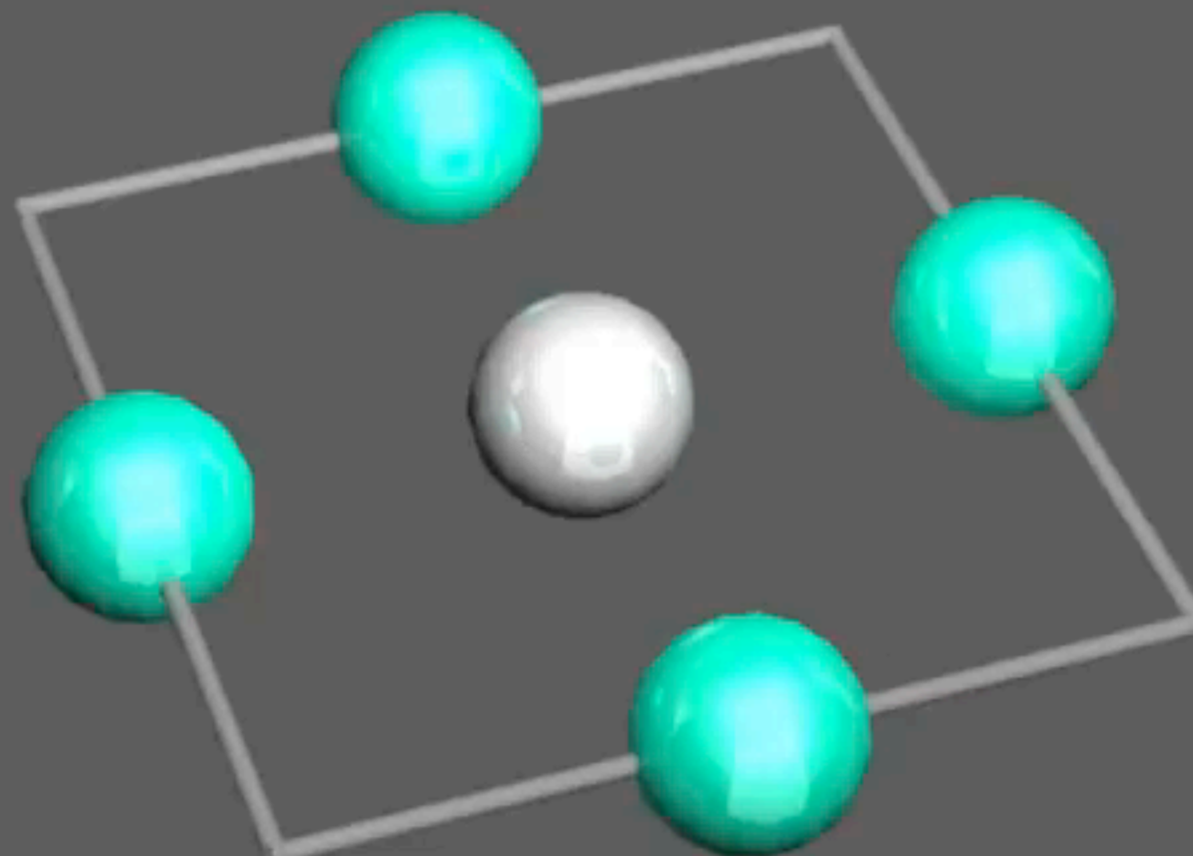
Surface Code Quantum Error Correction

Rodney Van Meter, Keio University
http://aqua.sfc.wide.ad.jp/

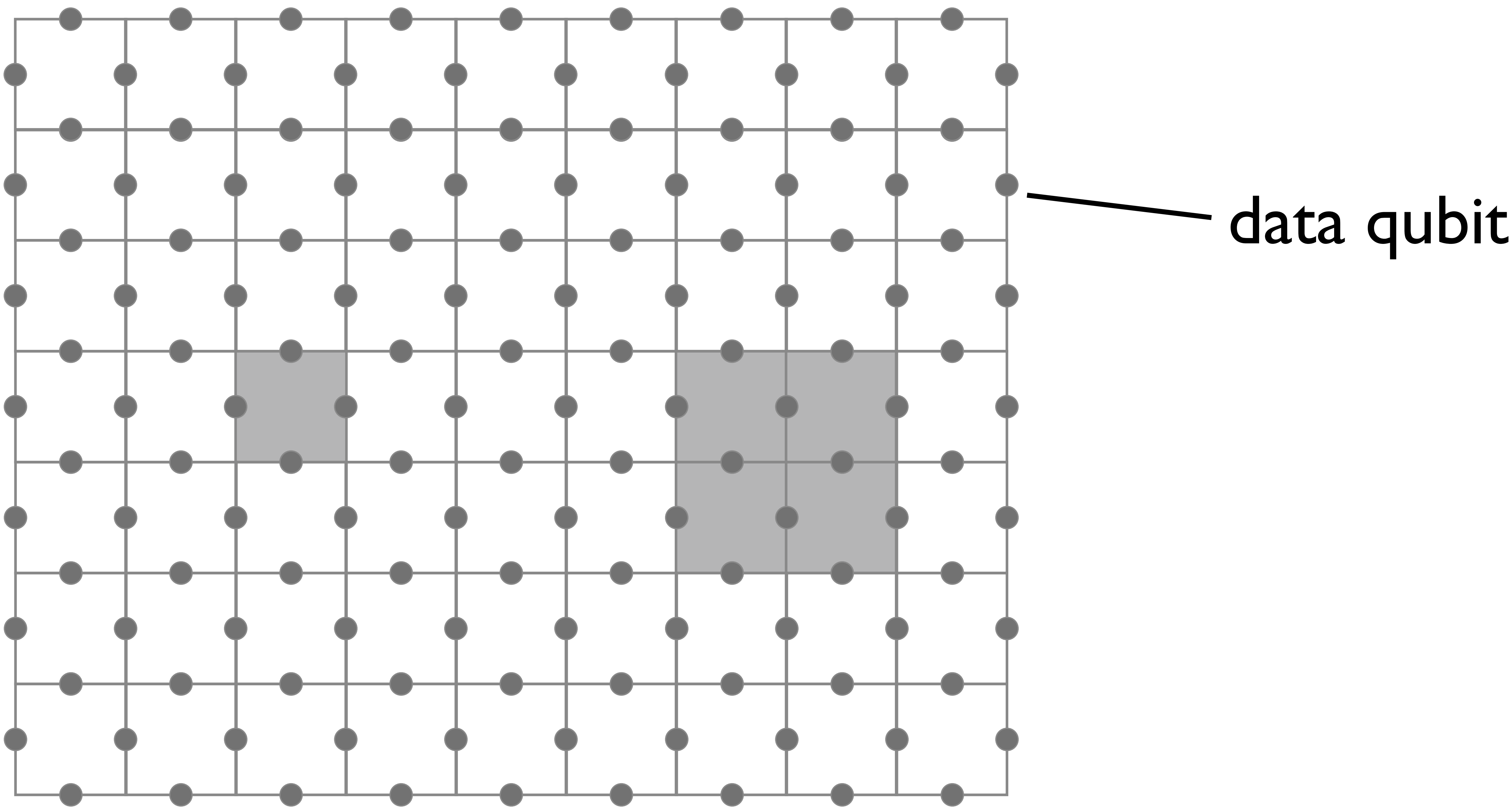
FIRST Project Summer School @ Kyoto U.

2011 Aug 16

(all the good slides and animations are by Shota Nagayama, Austin Fowler and Clare Horsman)

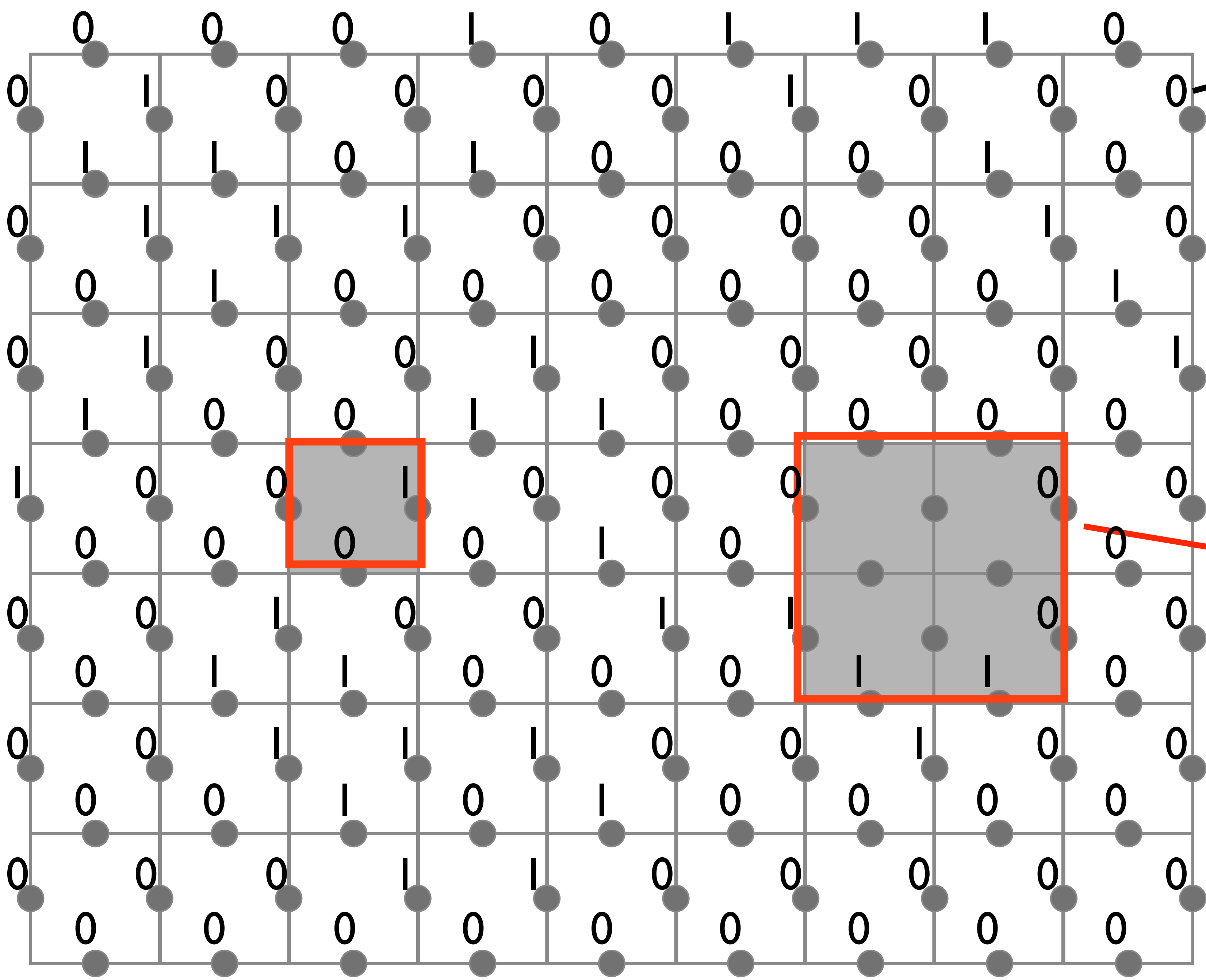


Simplest Example
~Encoding logical qubit~



3

Simplest Example
~Encoding logical qubit~



measured eigenvalue

data qubit

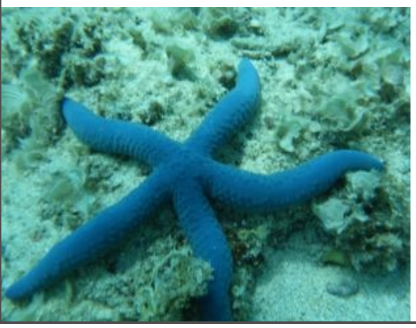
the edge of hole

odd parity: 0

even parity: 1

It's logical 0 here.

Surface Code Strengths



Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)

5

Surface Code Strengths

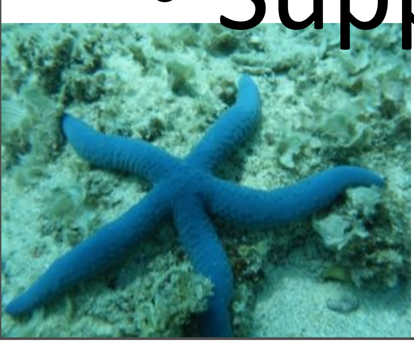
- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)
- High threshold: 1.4% for gate, memory, measurement errors

Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)
- High threshold: 1.4% for gate, memory, measurement errors
- Flexible:
  - –strength of EC grows incrementally (compare to concatenated CSS codes)
  - –software-assigned resources
  - –easy movement of logical qubits

Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)
- High threshold: 1.4% for gate, memory, measurement errors
- Flexible:
  - strength of EC grows incrementally (compare to concatenated CSS codes)
  - software-assigned resources
  - easy movement of logical qubits
- Supporting classical processing achievable

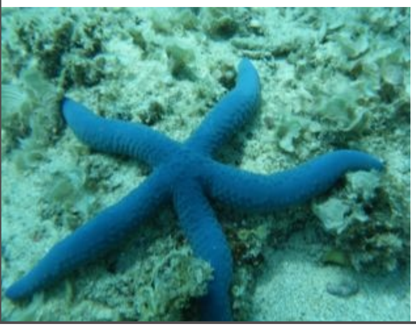


5

Scalability: fault-tolerance

- Trade-off between resources and threshold
- Thresholds
  - unlimited range, unlimited qubits: ~ 10^{-2}
    Knill, quant-ph/0410199
  - unlimited range, many qubits: ~ 10^{-3}–10^{-4}
    Steane, Phys. Rev. A 68, 042322 (2003)
  - 2D lattice, nearest neighbor: ~ 10^{-5}
    Svore, QIC 7, 297 (2007)
  - bilinear nearest neighbor: ~ 10^{-6}
    Stephens, QIC 8, 330 (2008)
  - linear nearest neighbor: ~ 10^{-8}
    Stephens, in preparation

Surface Code Drawbacks

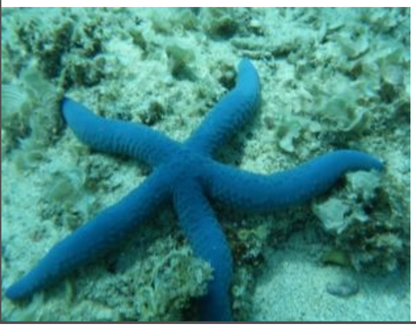


Surface Code Drawbacks

- Non-Clifford group operations difficult

Surface Code Drawbacks

- Non-Clifford group operations difficult
- Direct calculation of residual error rates difficult due to many error chains; determined via simulation

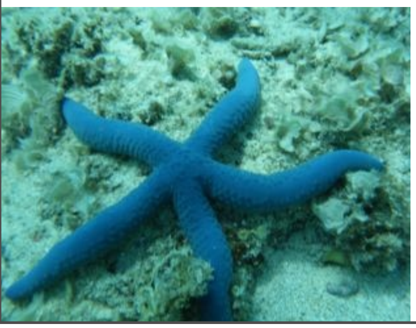


Surface Code Drawbacks

- Non-Clifford group operations difficult
- Direct calculation of residual error rates difficult due to many error chains; determined via simulation
- Almost uniform set of operations across whole device, but not quite!

Surface Code Drawbacks

- Non-Clifford group operations difficult
- Direct calculation of residual error rates difficult due to many error chains; determined via simulation
- Almost uniform set of operations across whole device, but not quite!
- *Extremely* difficult to explain to classical computer engineers!



Outline

- Stabilizers
- Surface Code Operation
- Theory of the Surface Code
- Advanced topics
- (system architecture reserved for next lecture)

Stabilizers

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv +1 \text{ eigenstate of } Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$|+\rangle \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \equiv +1 \text{ eigenstate of } X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$|0\rangle \quad \text{or} \quad Z$$

$$|+\rangle \quad \text{or} \quad X$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad \text{or} \quad Z_1 Z_2,\ X_1 X_2$$

- n qubits, n independent commuting stabilizers $\Rightarrow$ unique state

Stabilizers

- *n-k* stabilizers on set of *n* qubits leaves *k* degrees of freedom, can encode *k* logical qubits
- Measure set of stabilizers to get error syndrome
- See Gottesman PhD. thesis, and the set of notes by Clare Horsman in download materials

Introduction to stabilizer theory

June 16, 2011

1 Stabilizer definition

In quantum mechanics, a **state** is given by a **vector**, and an **operator** is given by a **matrix**. The state of N qubits is a 1×2^N vector, and an operation on the state is a $2^N \times 2^N$ matrix.

For example, a state of 1 qubit could be

$$|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

and an operation on it could be

$$X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

In linear algebra, if for a matrix M there is a **vector V** and a **scalar** v such that

$$\mathrm{M}.\mathbf{V} = v.\mathbf{V}$$

then **V** is an **eigenvector** of M and v is the associated **eigenvalue**.

The vector **V** can be the eigenvector of more than one matrix. The vector **V** can be **fully defined** by the set $\{\mathrm{M}, v\}$ of matrices that **V** is an eigenvector of.

The **stabilizers** of a state $|\psi\rangle$ are the operators $\frac{1}{v}.\mathrm{M}$ where $|\psi\rangle$ is an eigenvector of M with eigenvalue v.

We see that

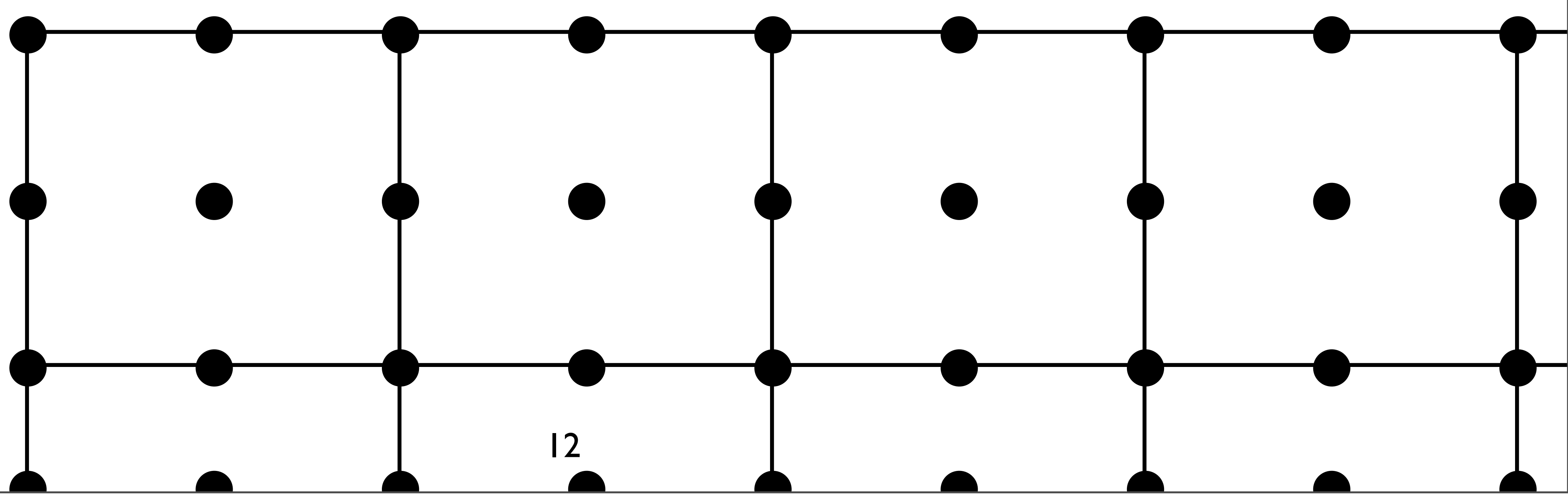
$$X|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle$$

so X is a stabilizer of $|+\rangle$.

Operations

- Lattice cycle
- Detecting & correcting errors
- Holes as logical qubits
- Single-qubit X and Z gates
- Moving holes
- Braiding for CNOT (Primal & dual lattice)
- Non-Clifford gates using singular qubits
- “Singular factories”
- Measuring a logical qubit

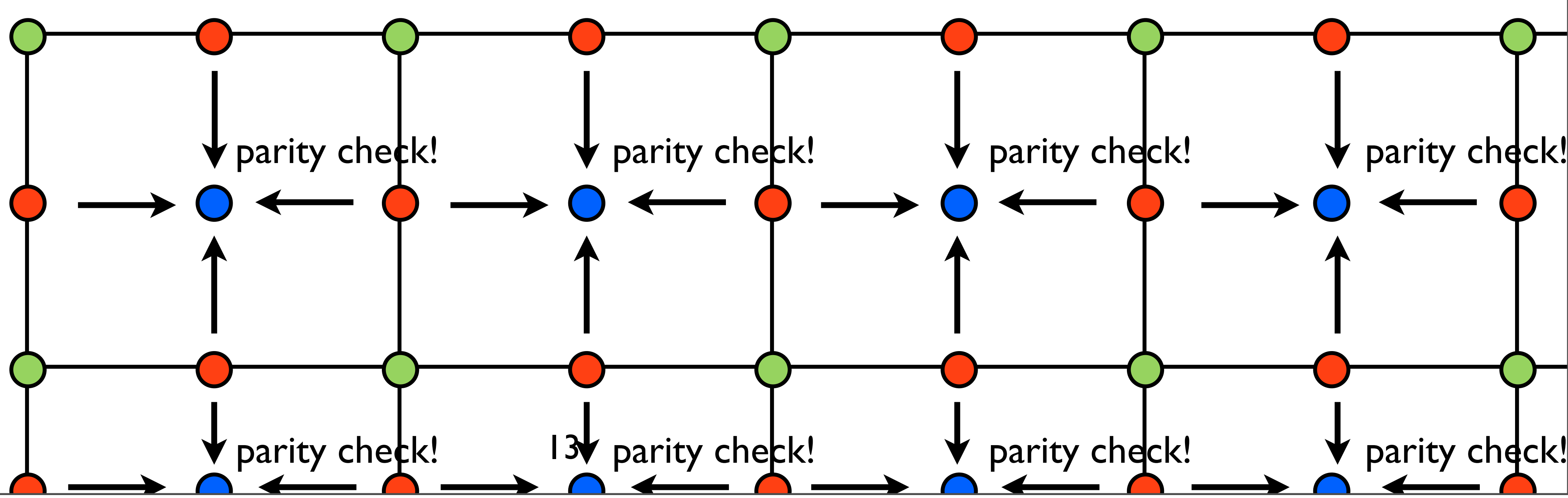
Surface Code



Surface Code

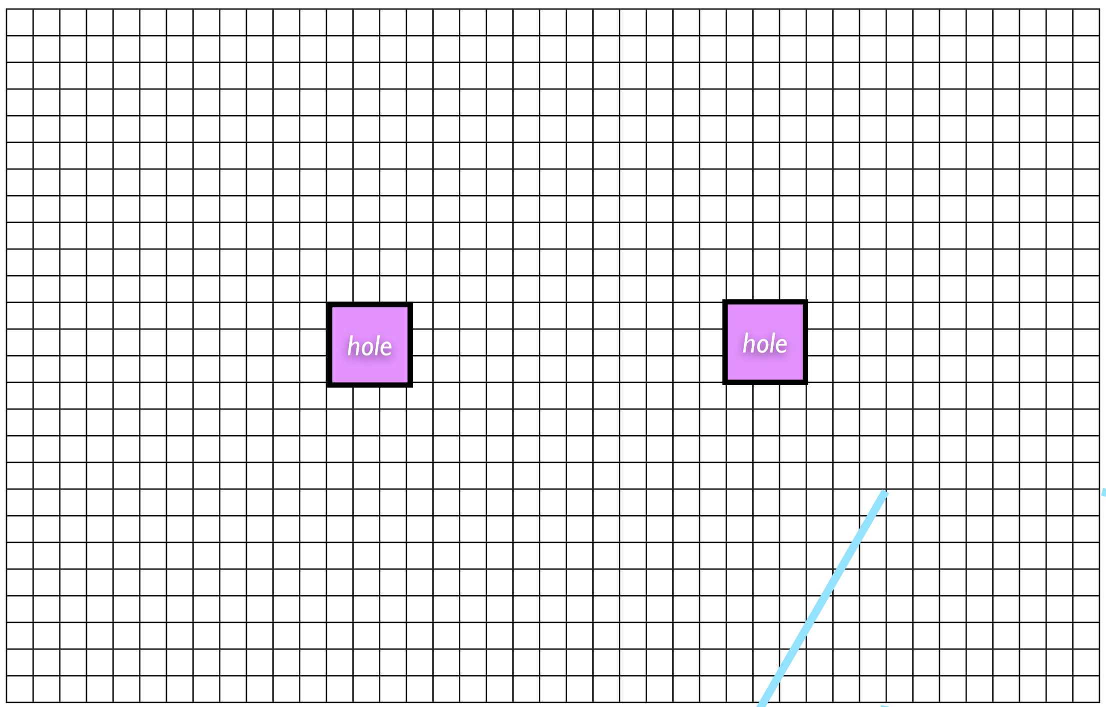
edges of two holes =logical qubit

- data qubit
- phase-flip checker
- bit-flip checker



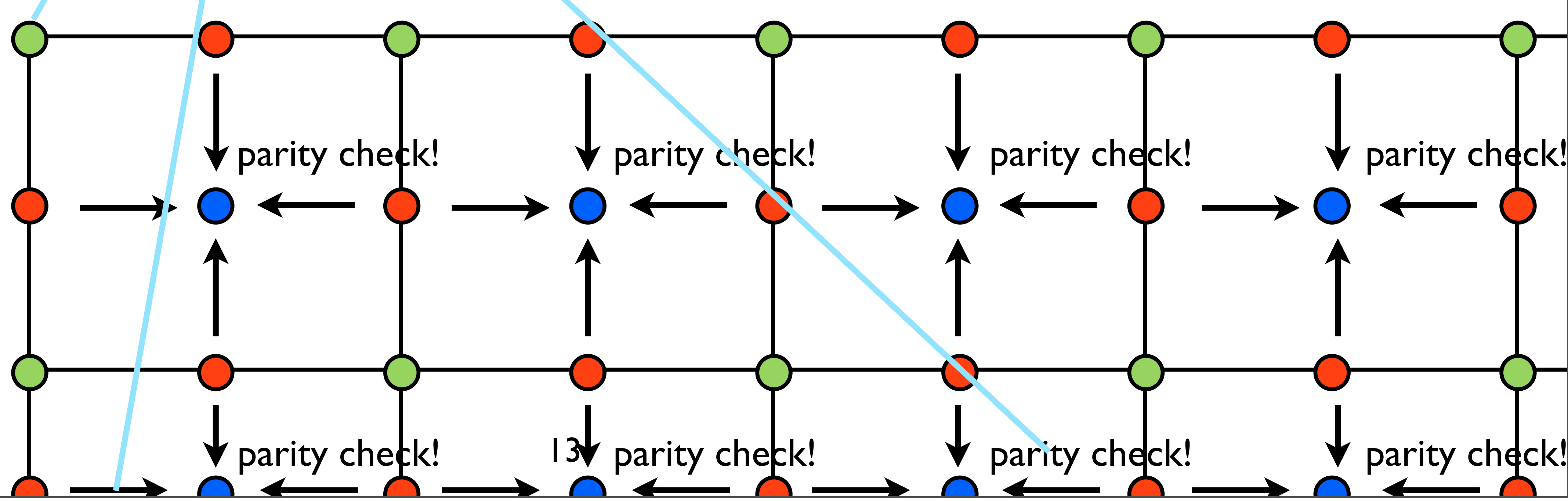
13

Surface Code

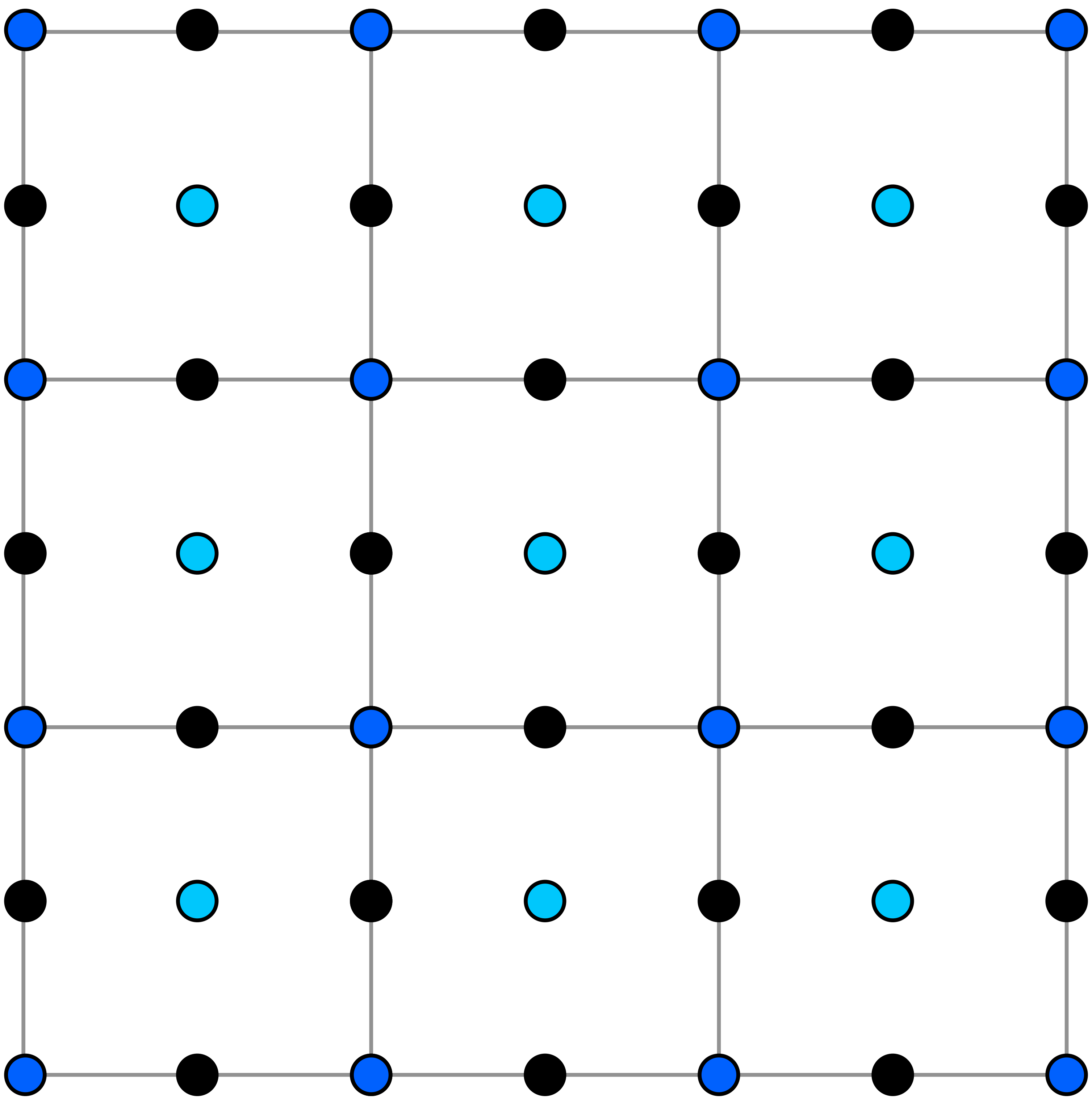


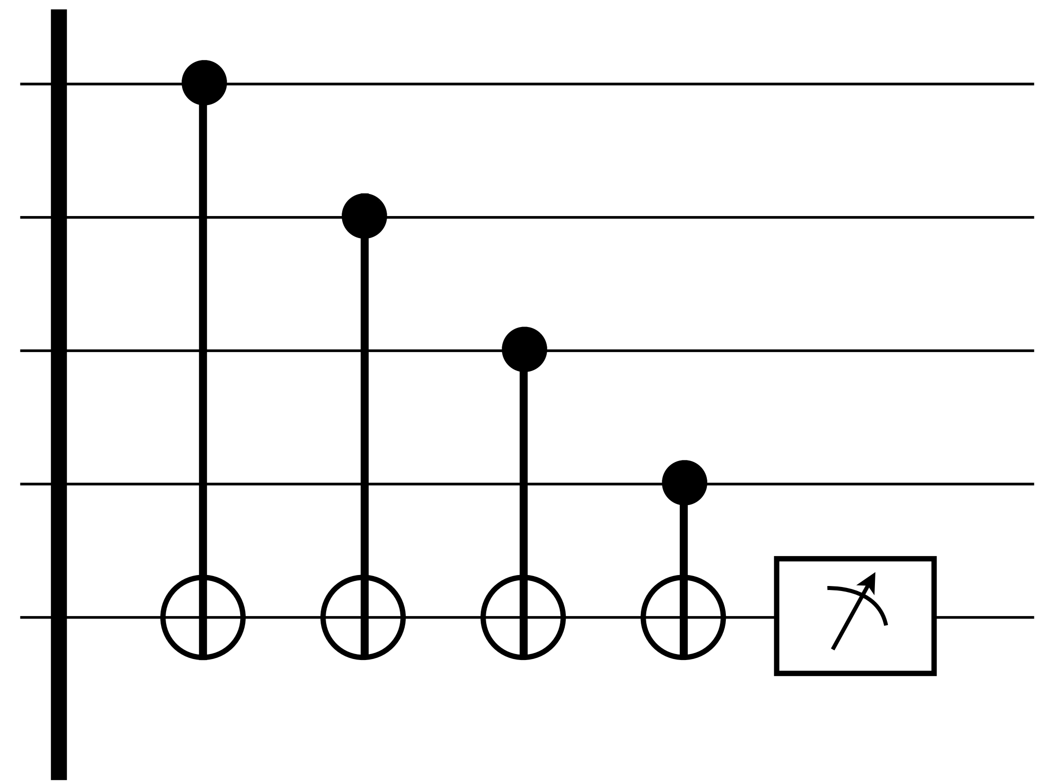
edges of two holes =logical qubit

- data qubit
- phase-flip checker
- bit-flip checker

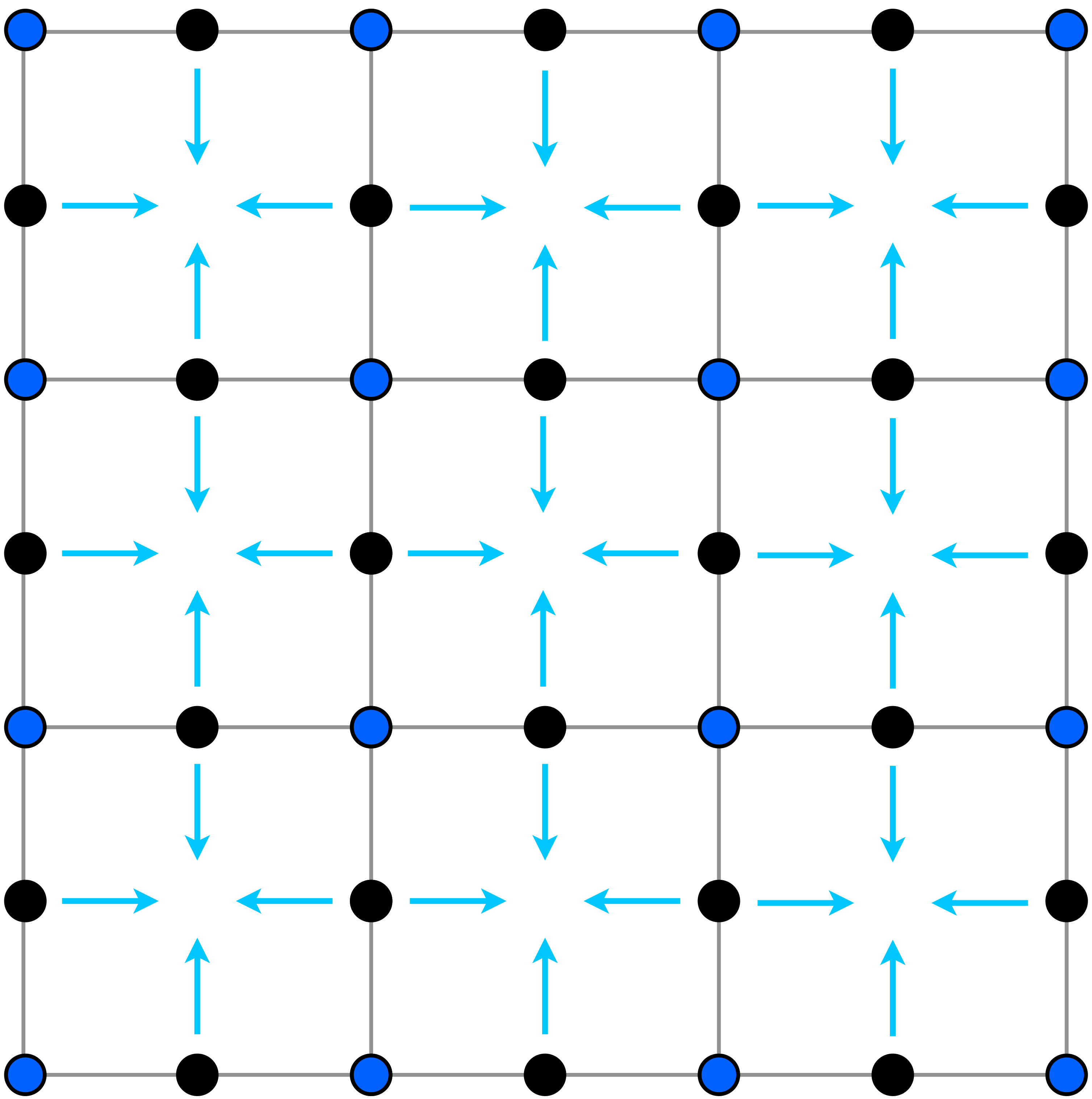


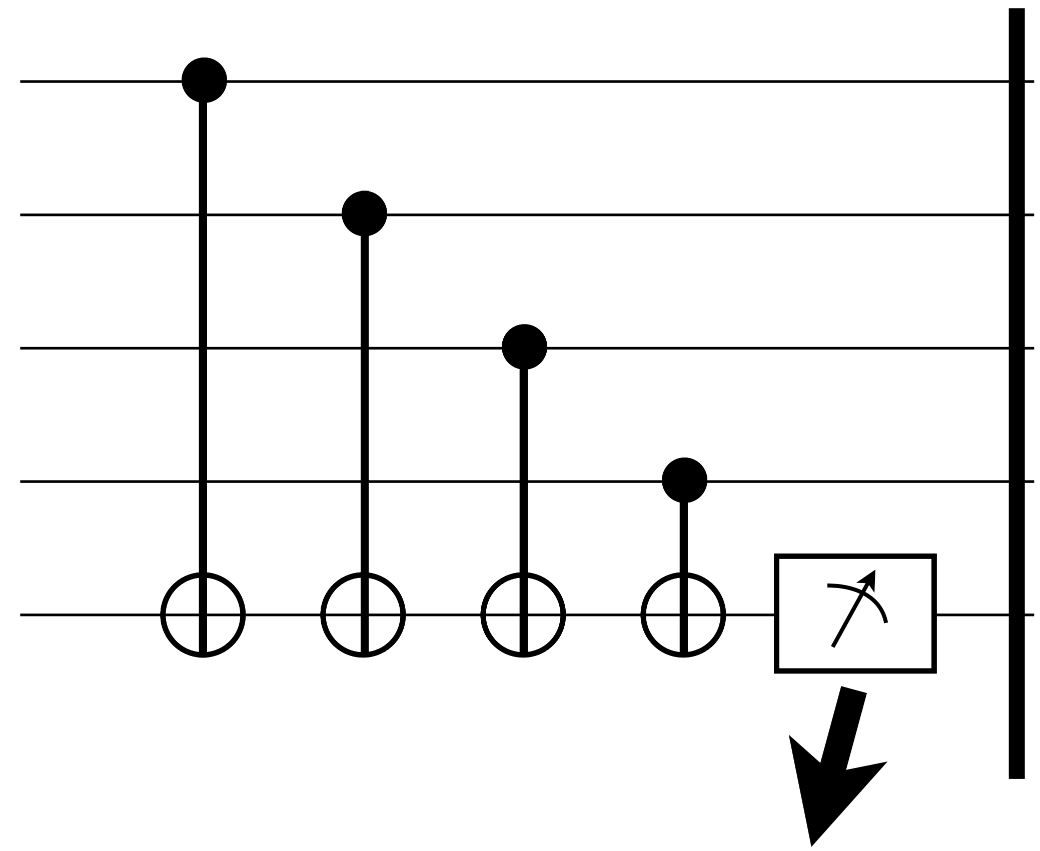
Entangling the Lattice





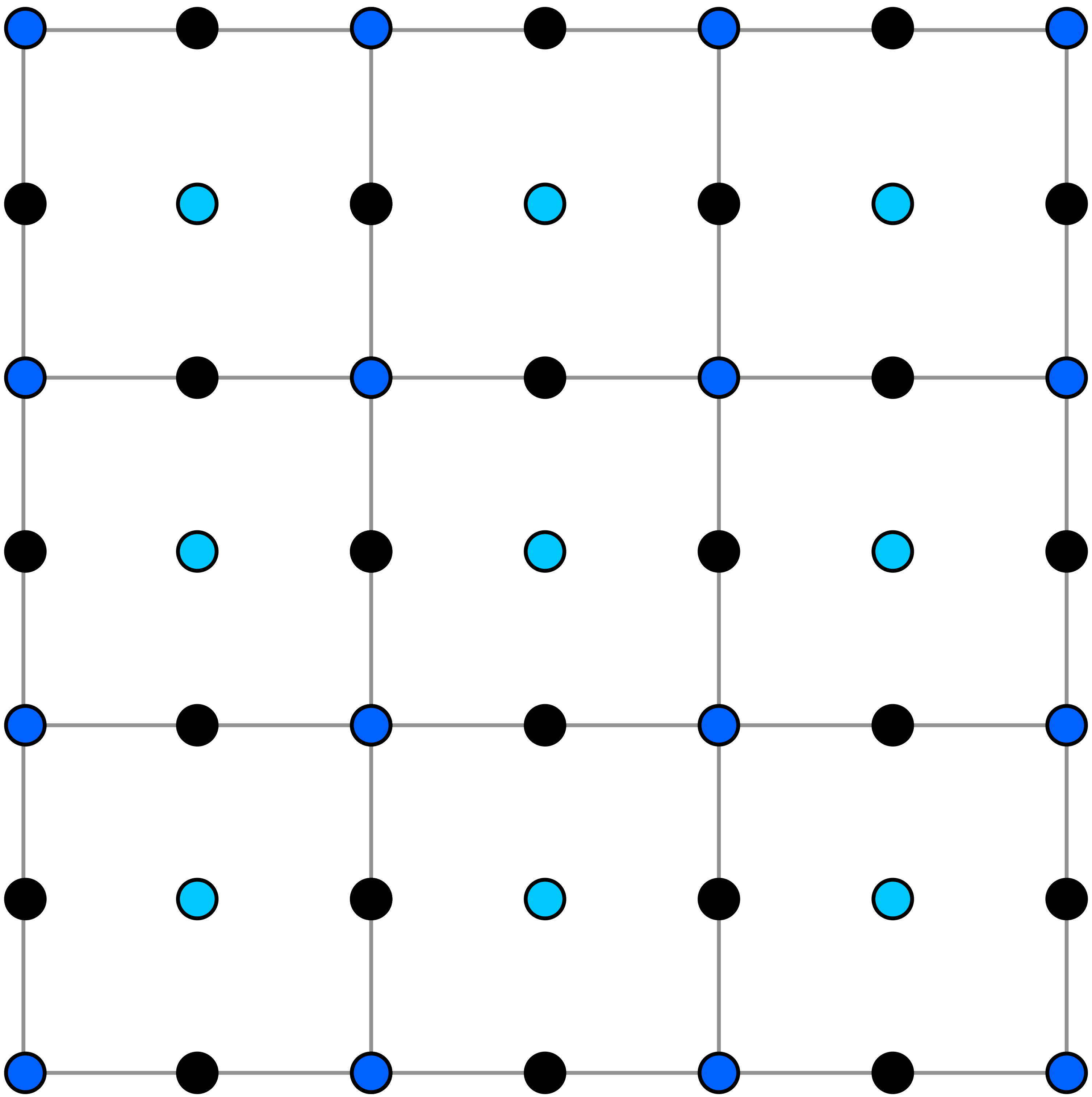
Entangling the Lattice

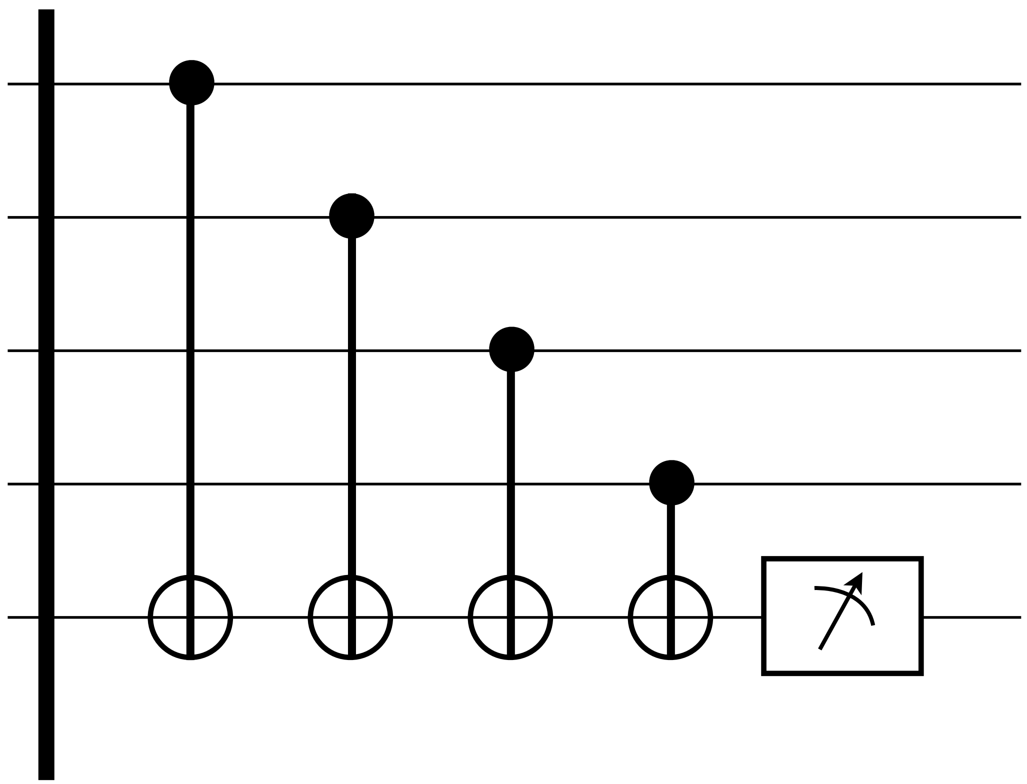




0	0	0
0	0	0
0	0	0

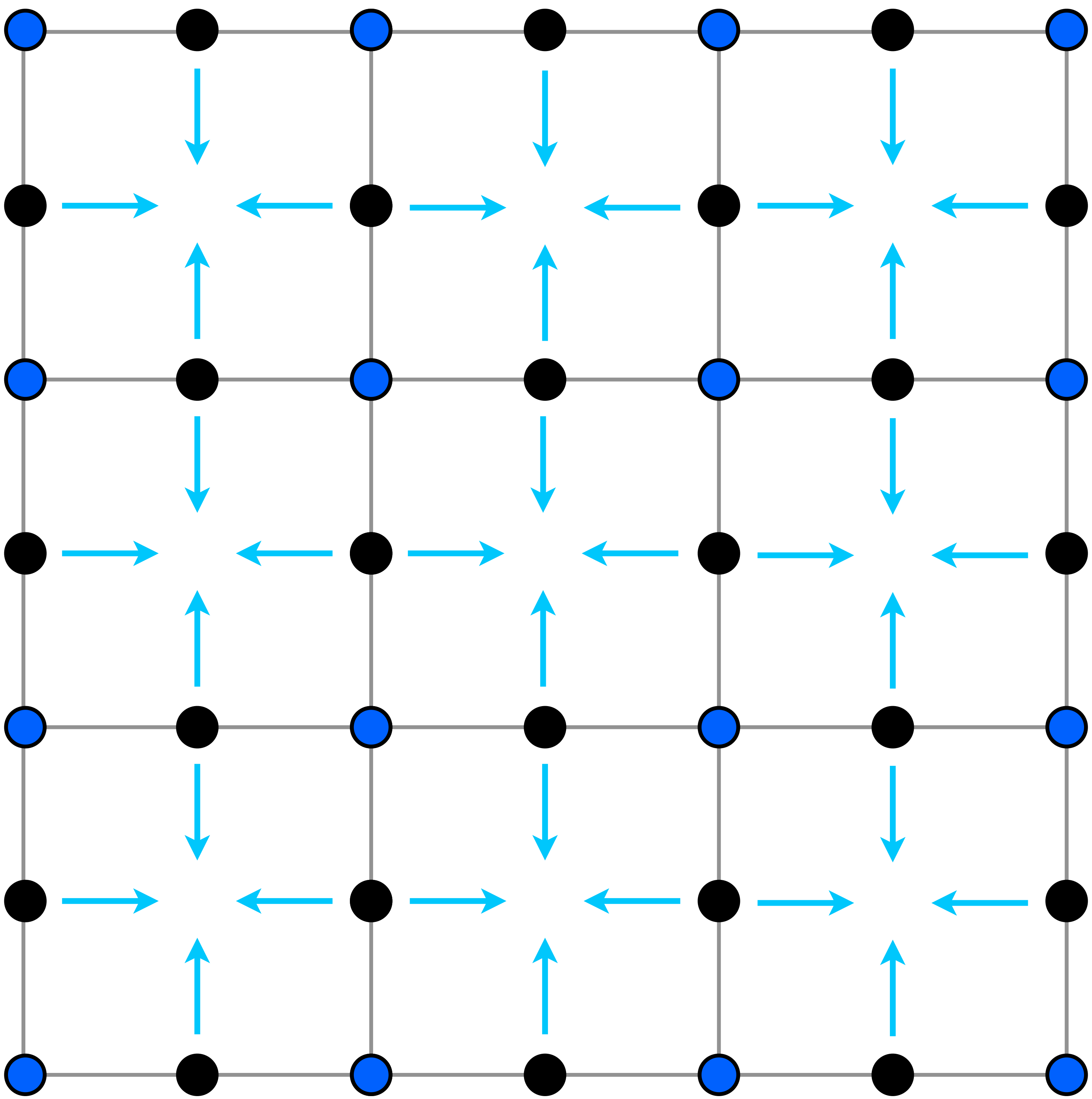
Measuring Z stabilizers

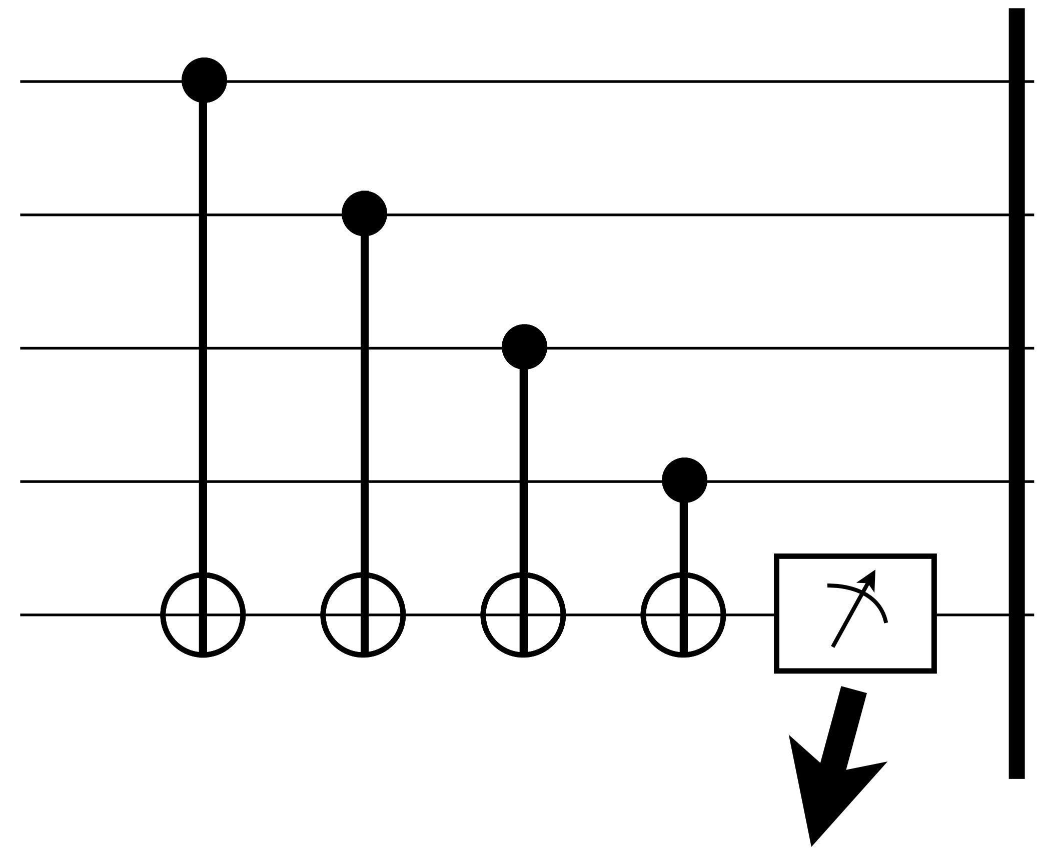




15

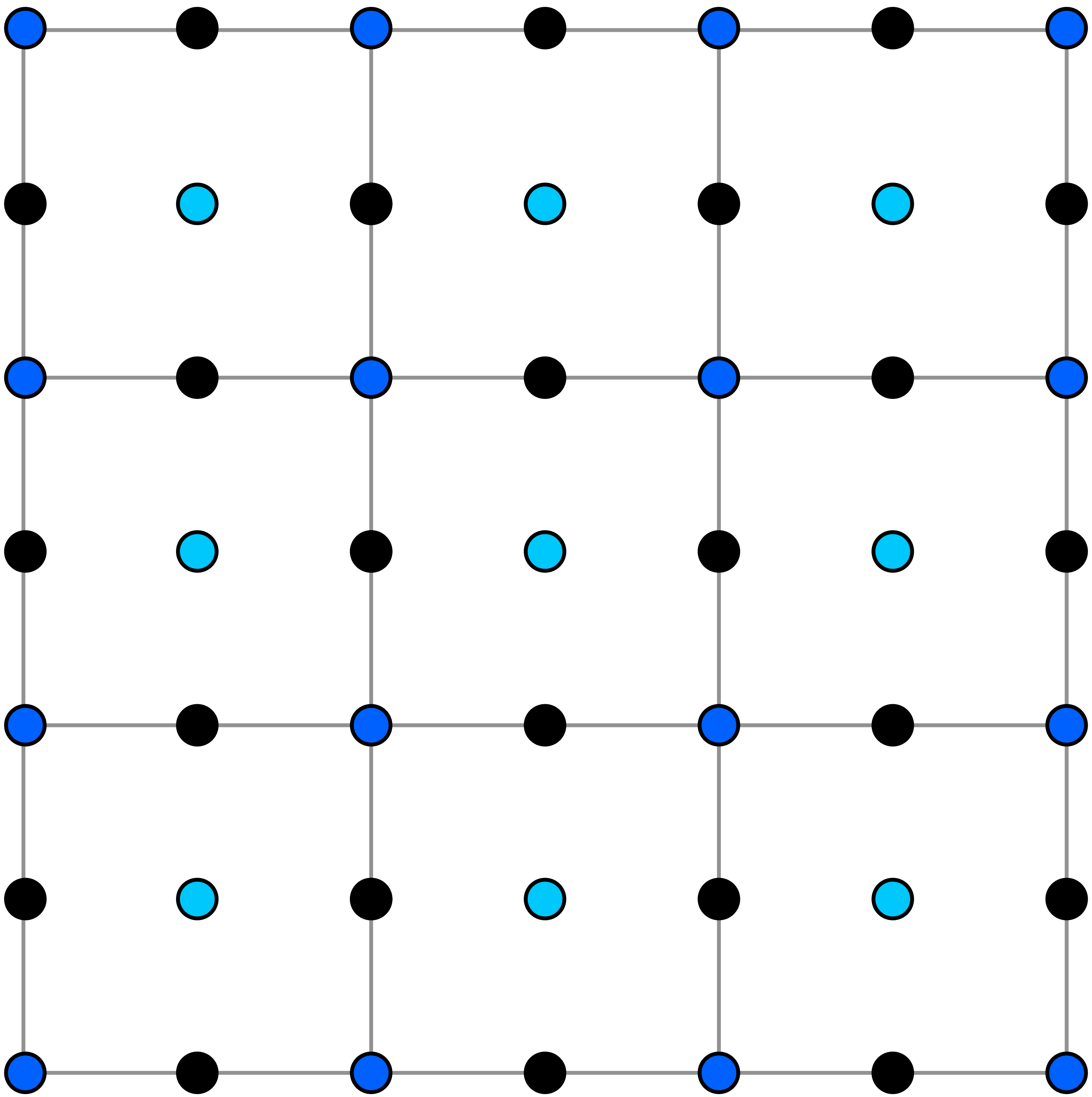
Measuring Z stabilizers

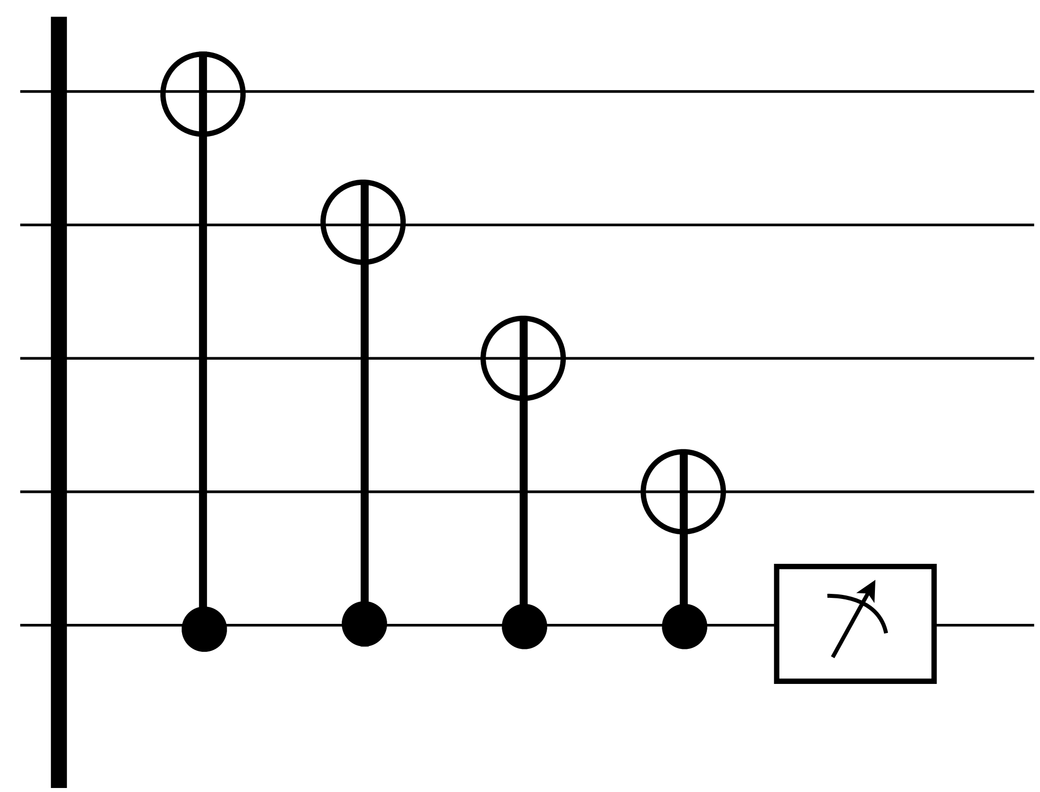




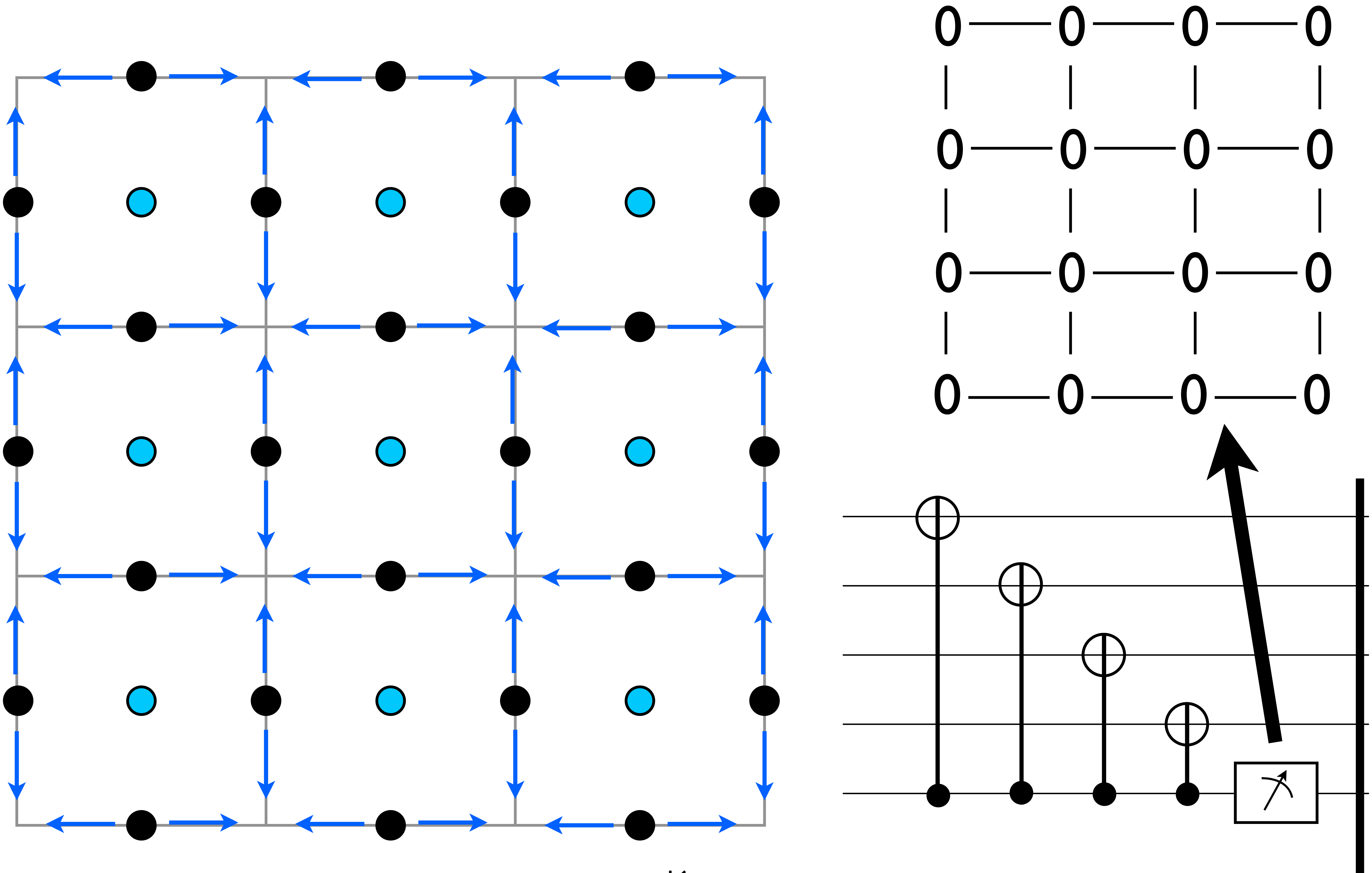
0	0	0
0	0	0
0	0	0

Measuring X stabilizers

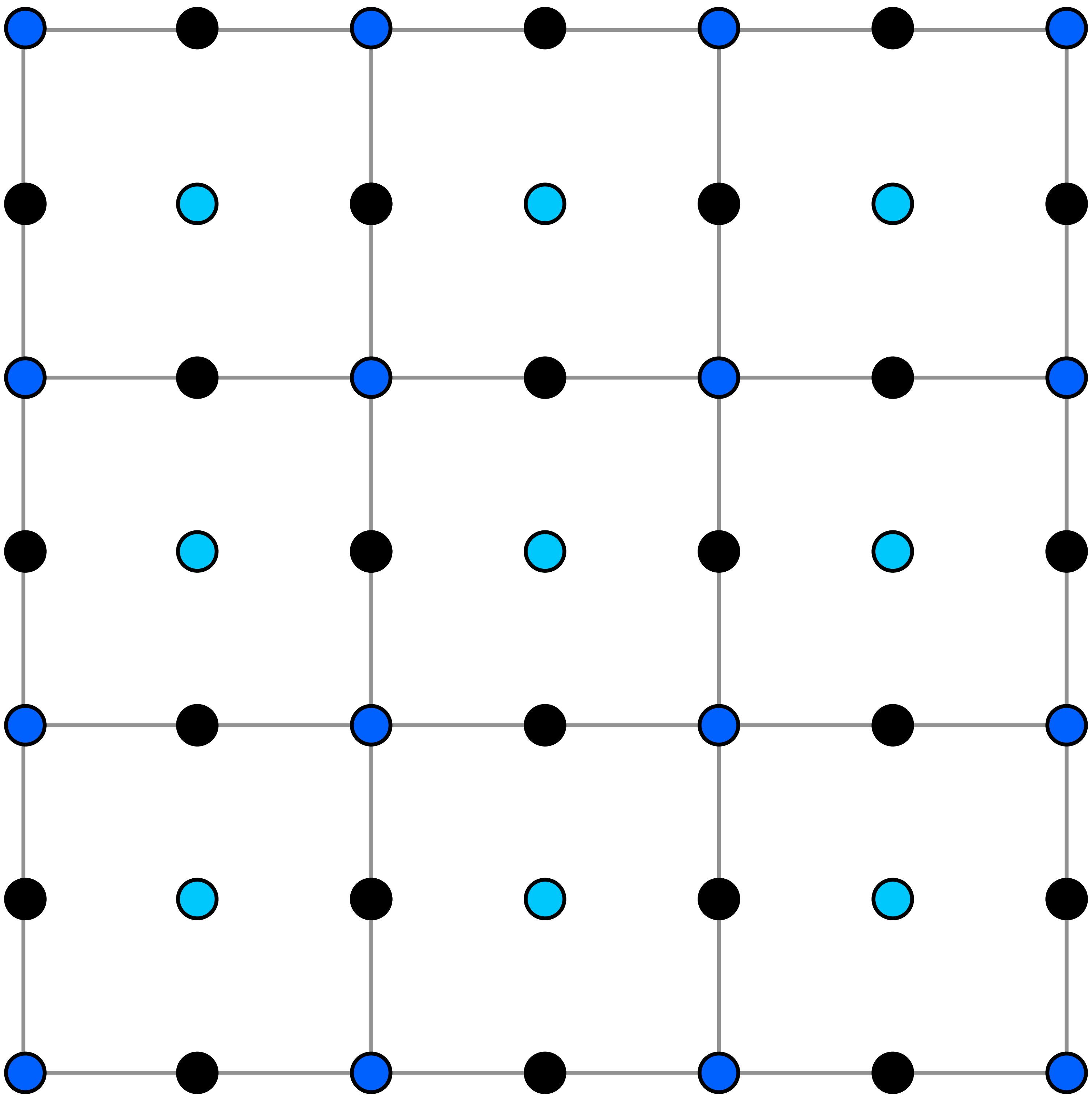


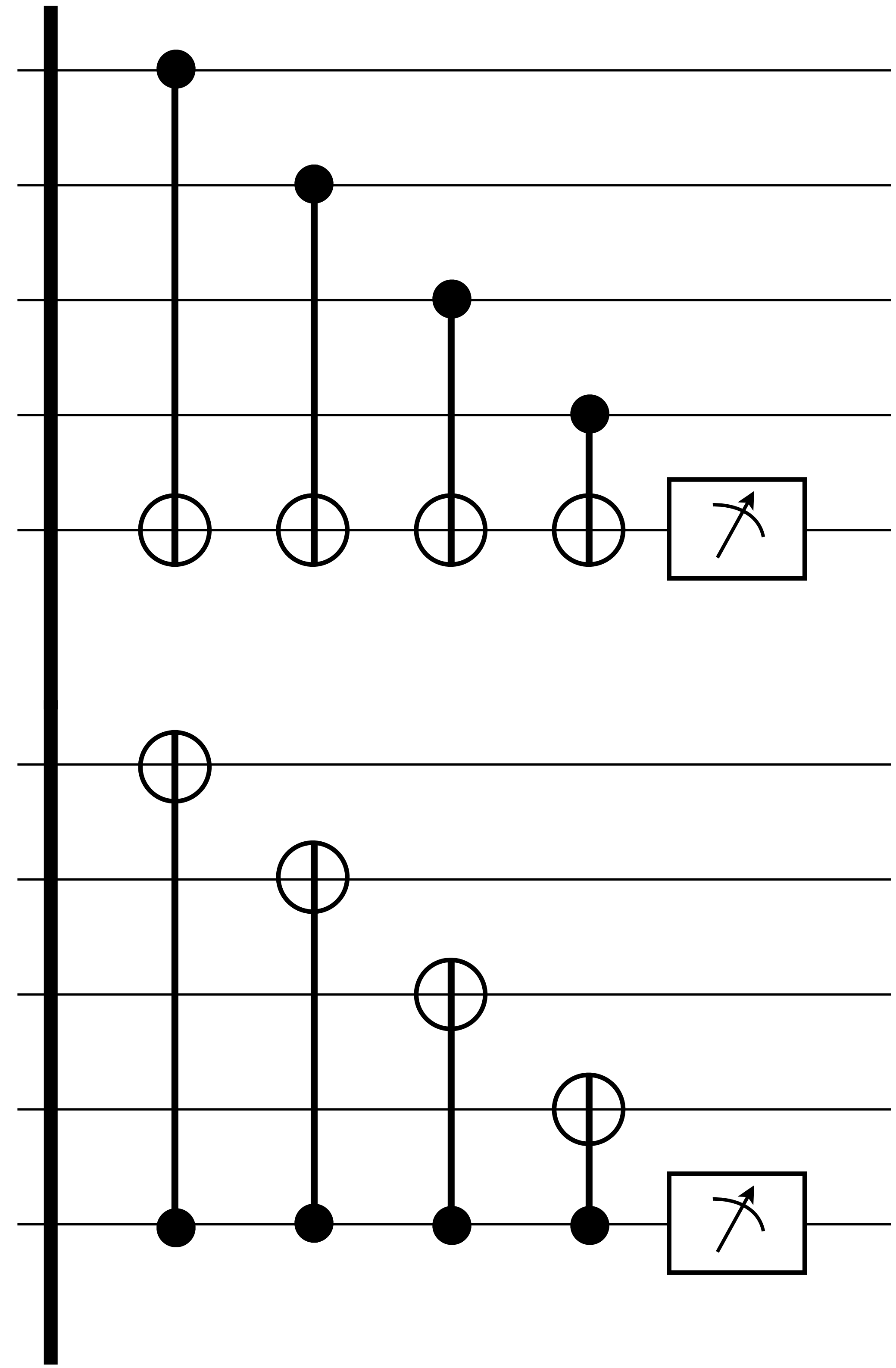


Measuring X stabilizers

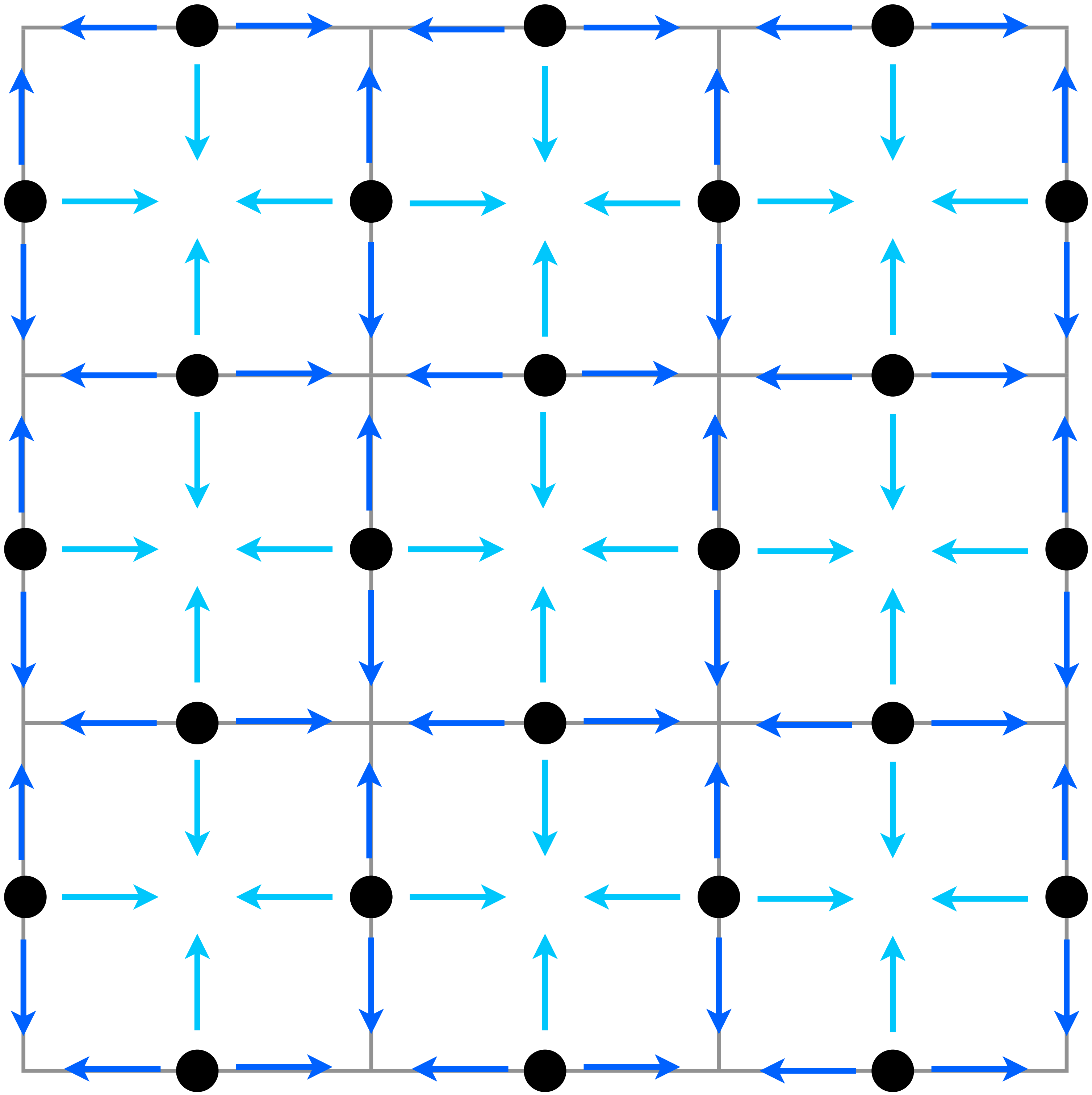


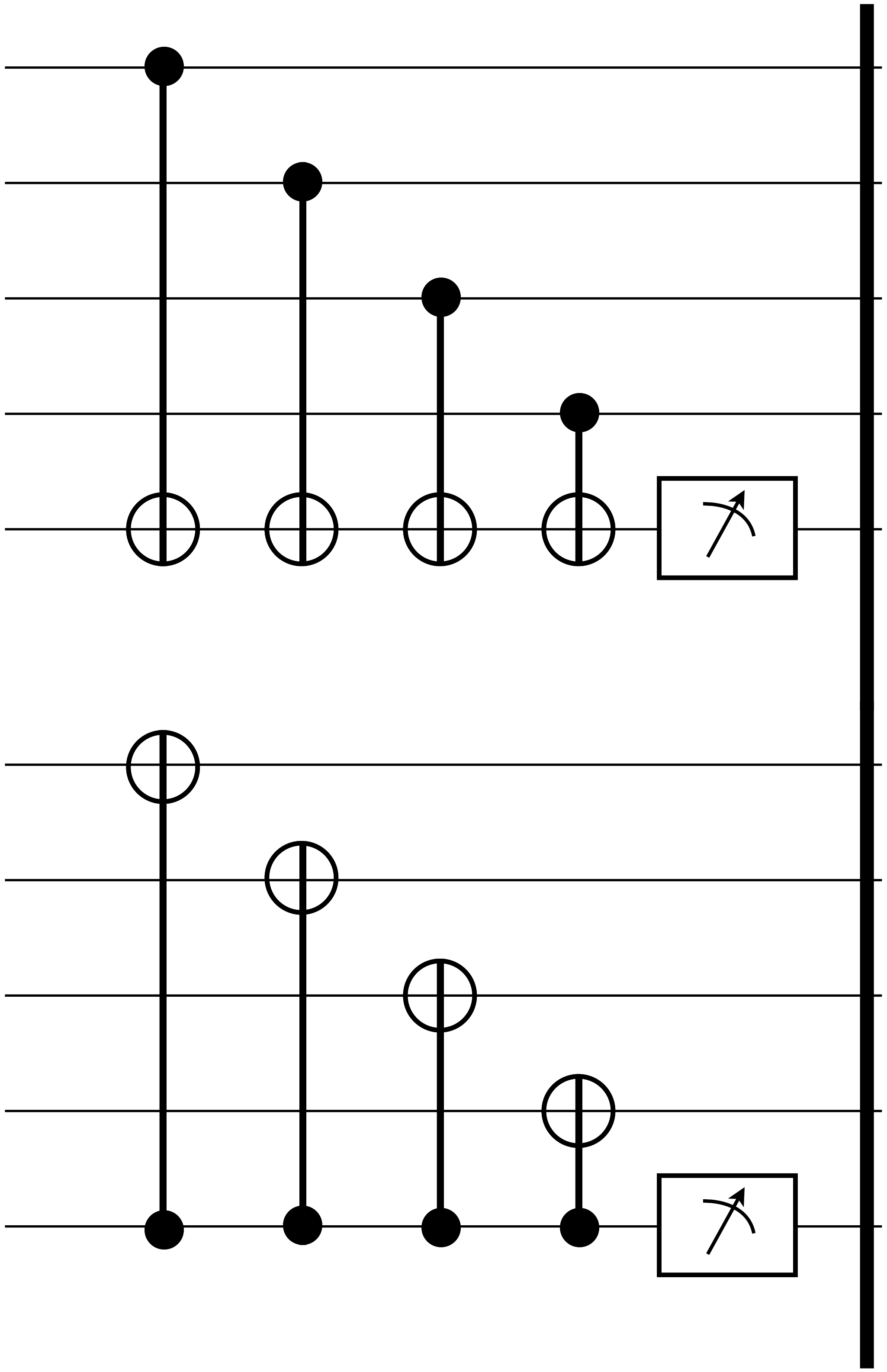
Full lattice cycle





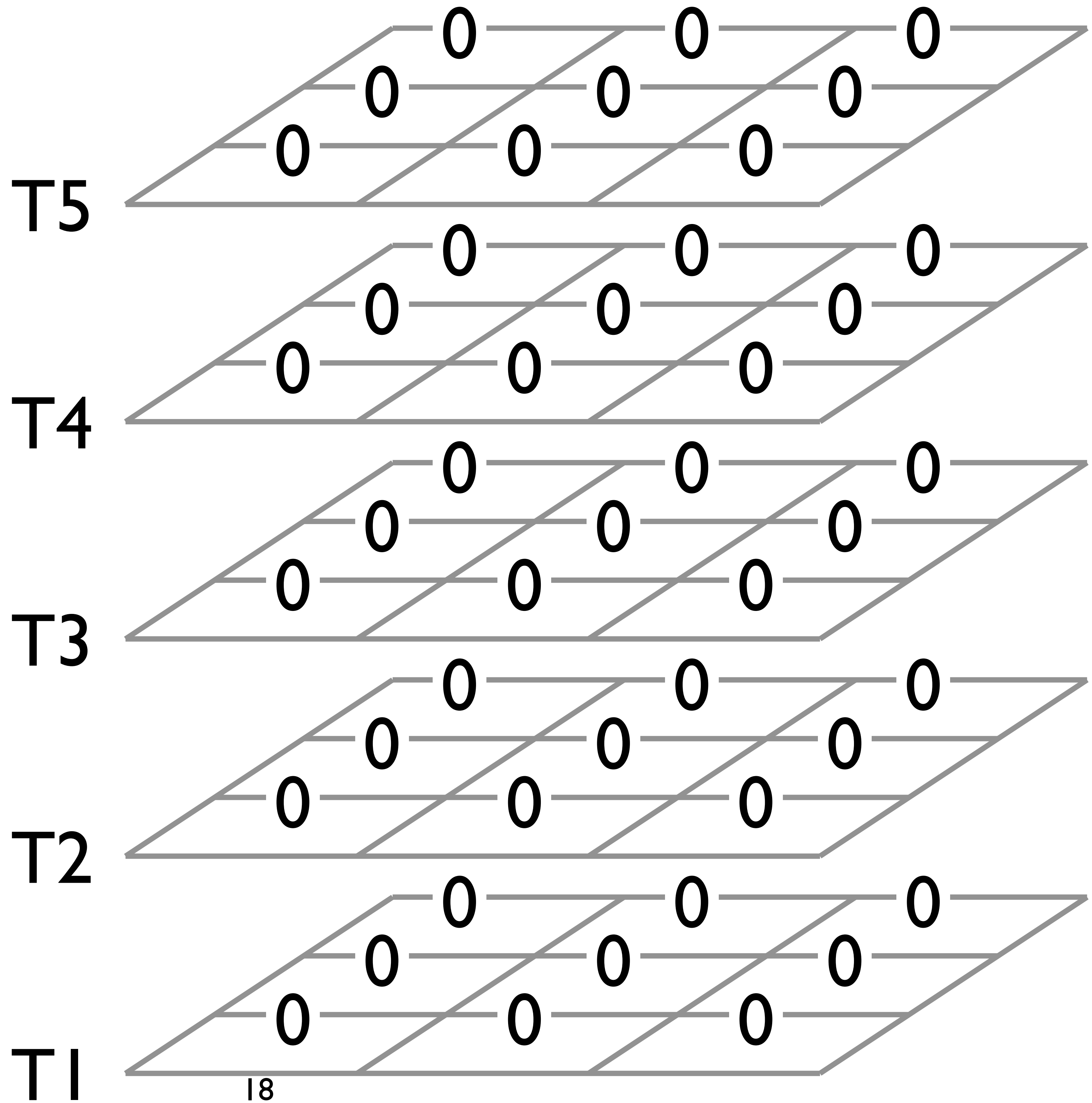
Full lattice cycle



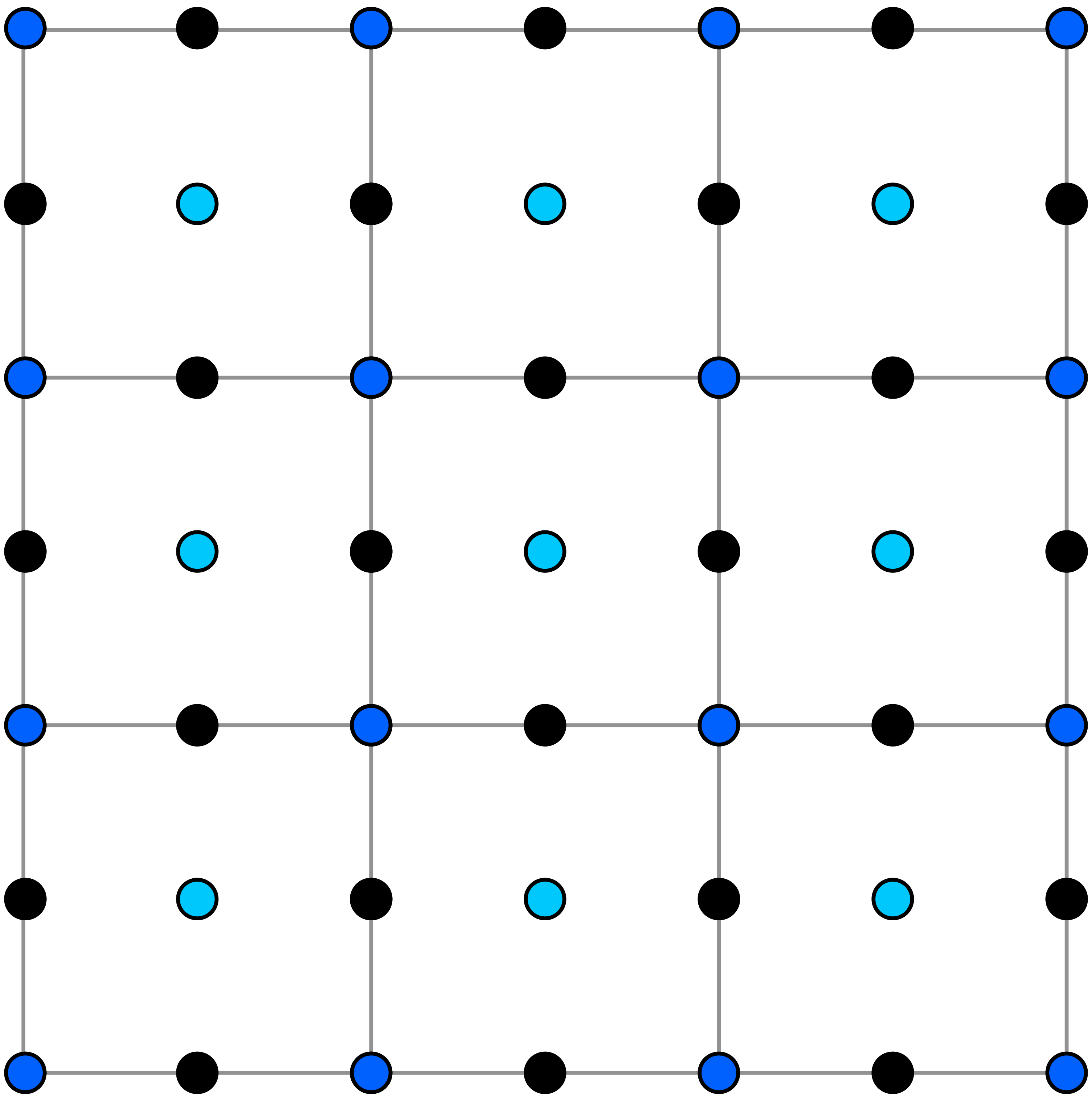


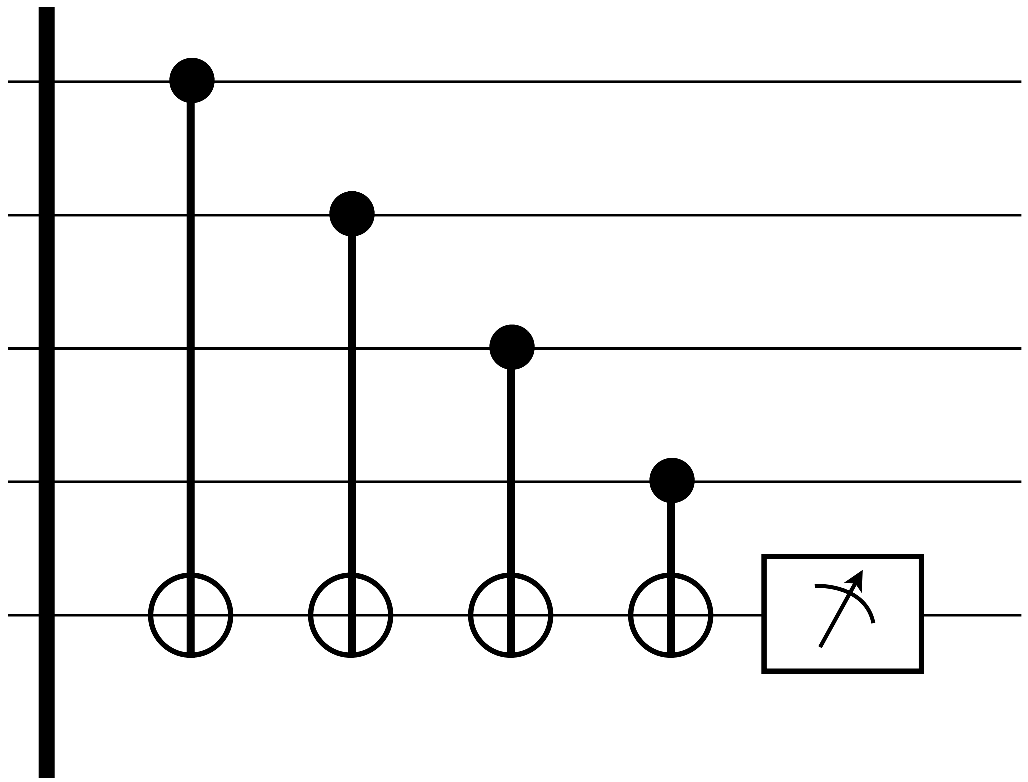
17

Error syndromes

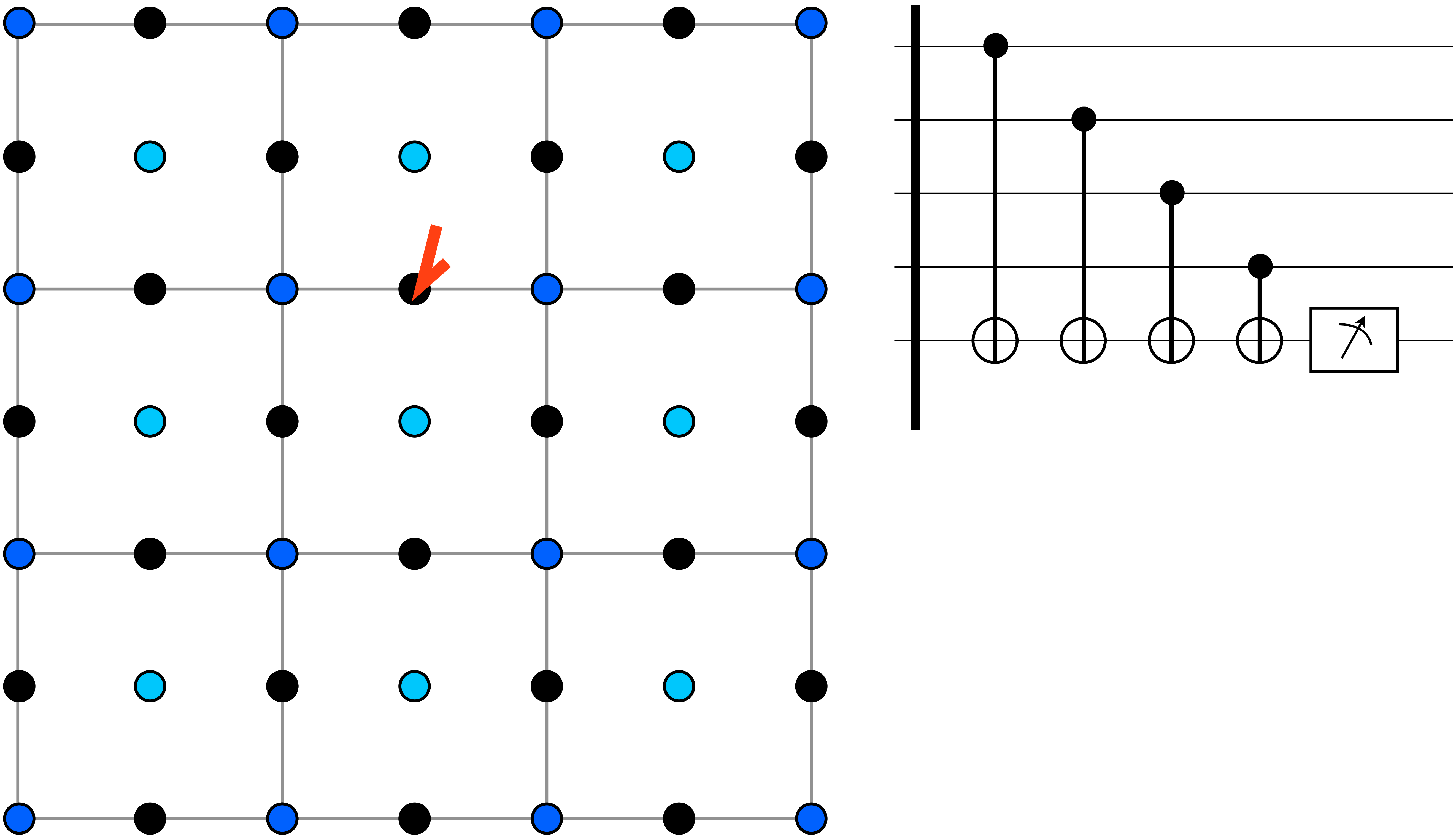


Memory Error 1

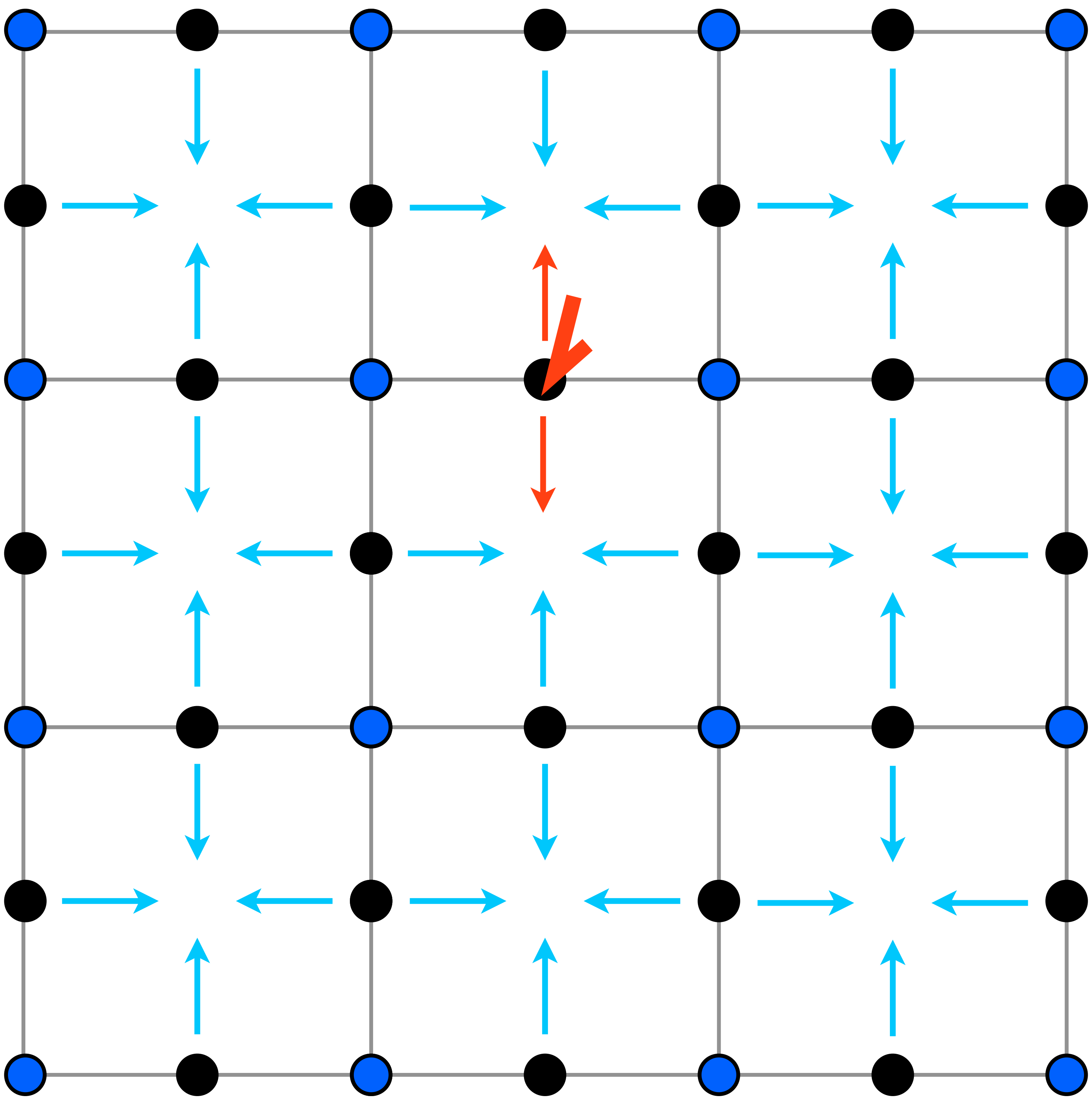


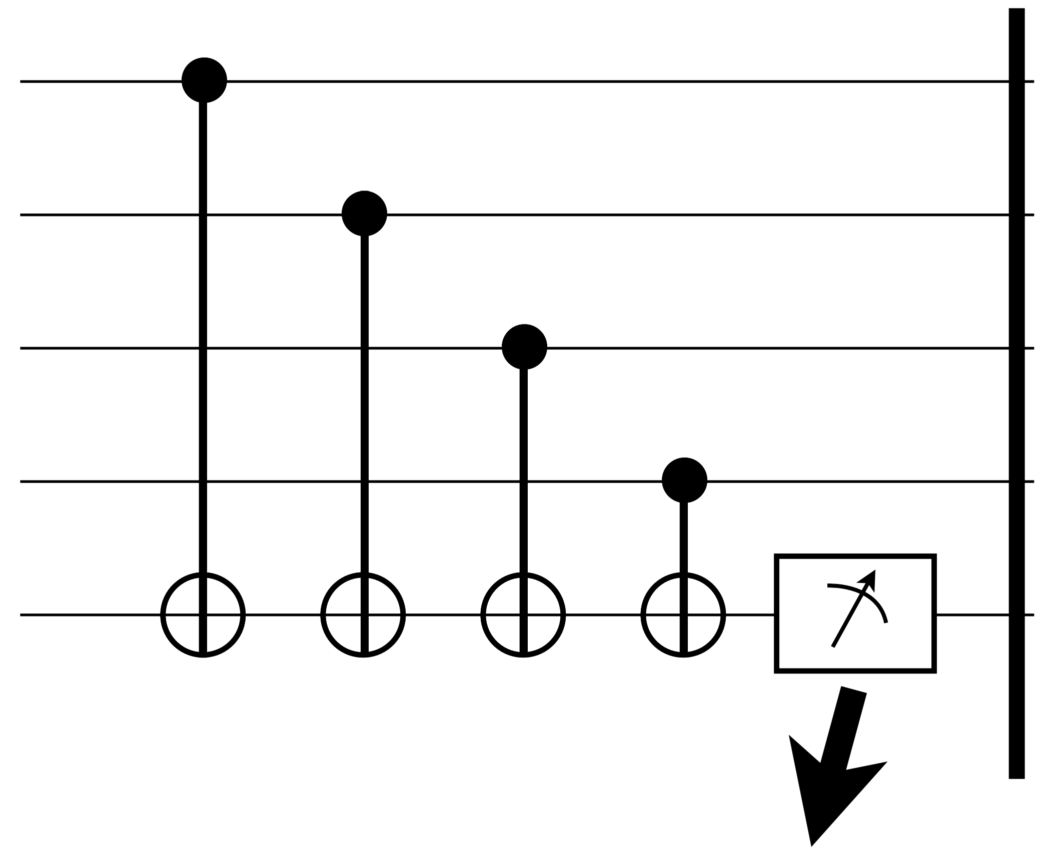


Memory Error 1



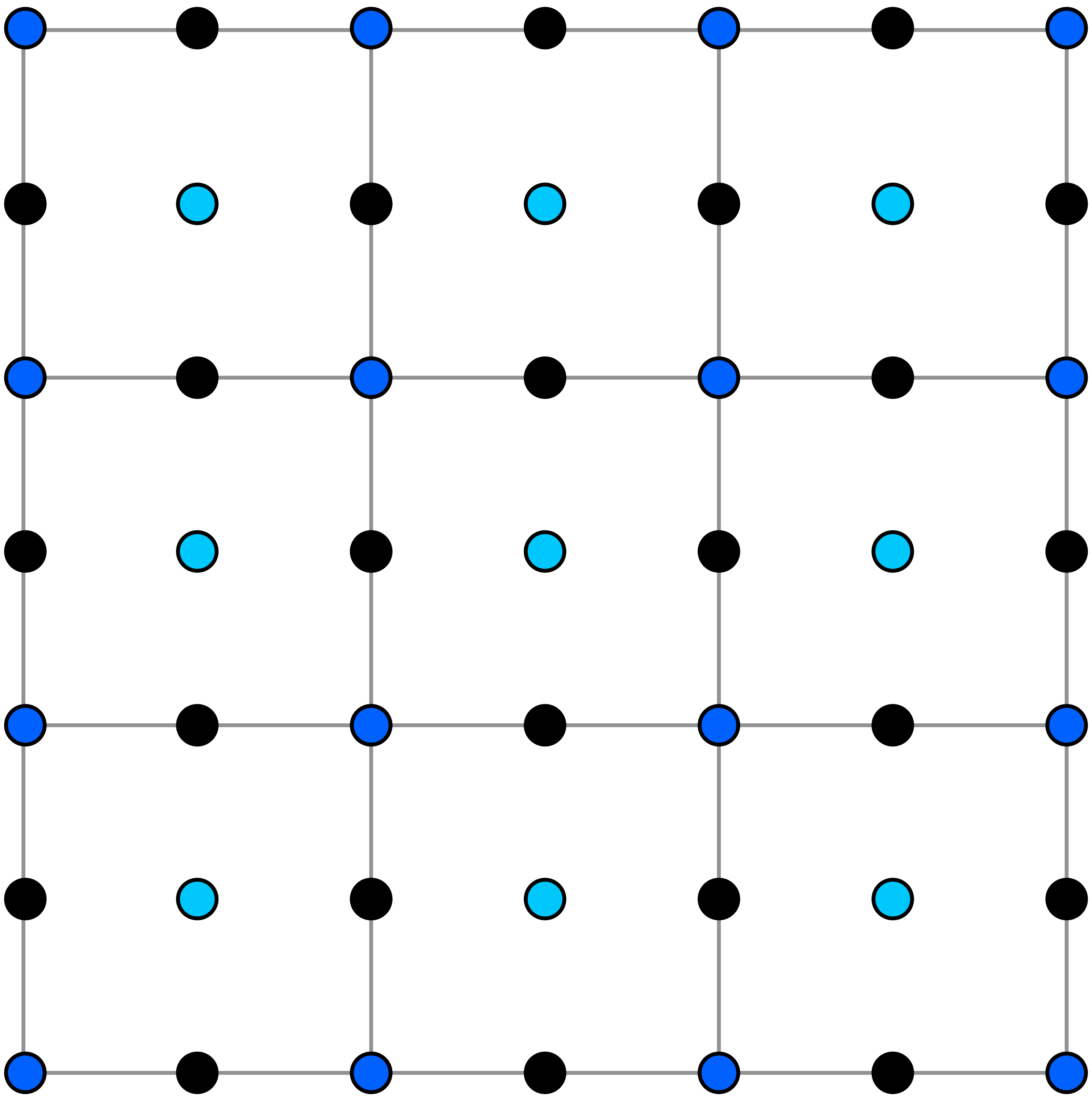
Memory Error 1

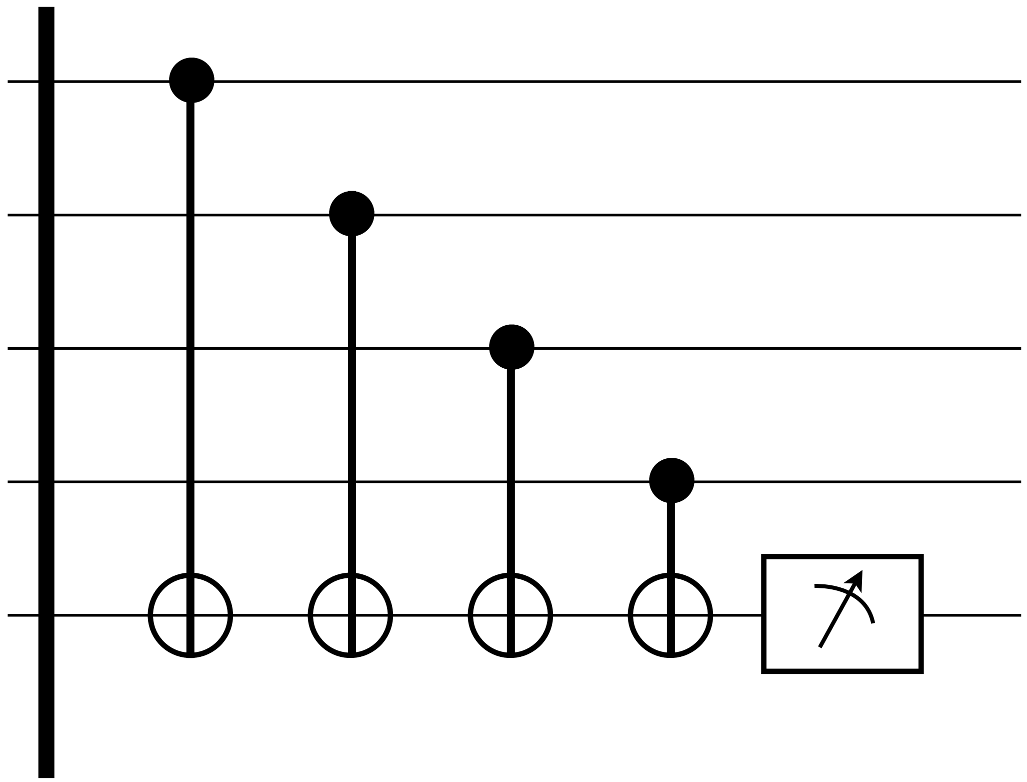




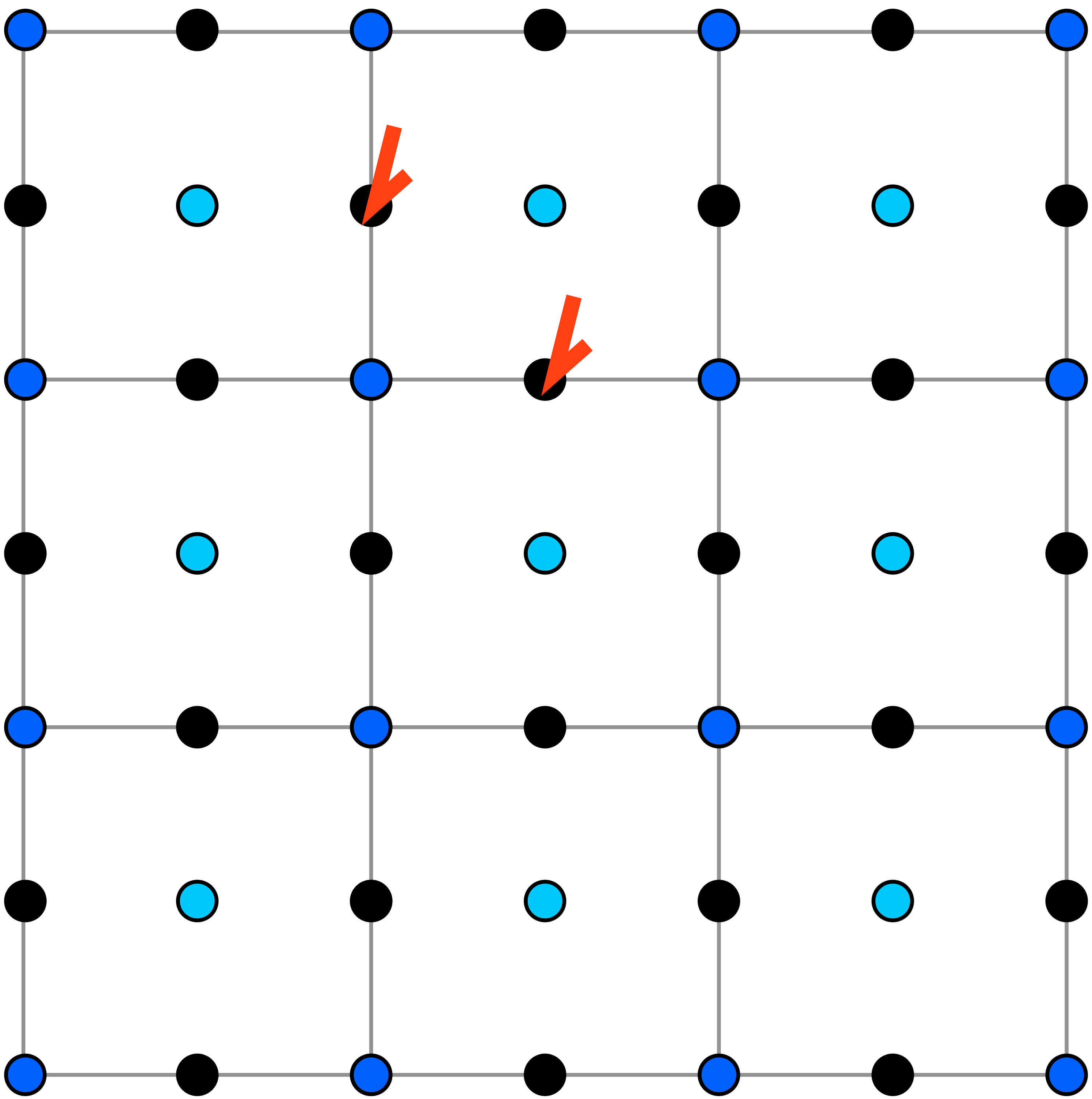
0	1	0
0	1	0
0	0	0

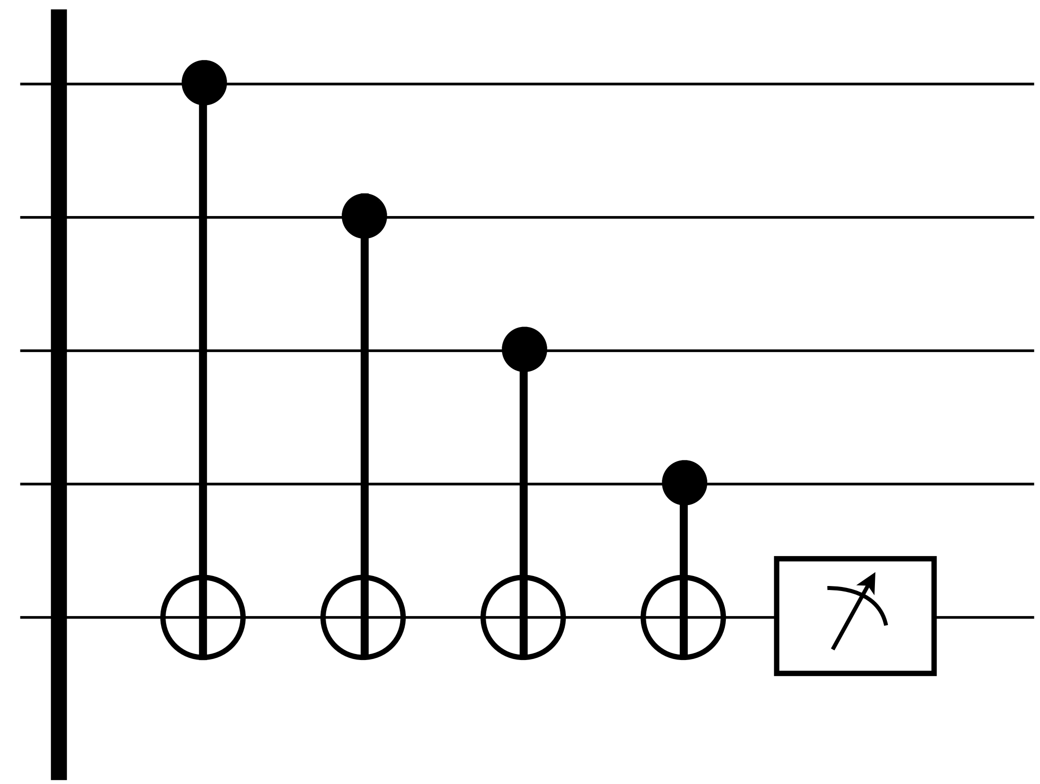
Memory Error 2



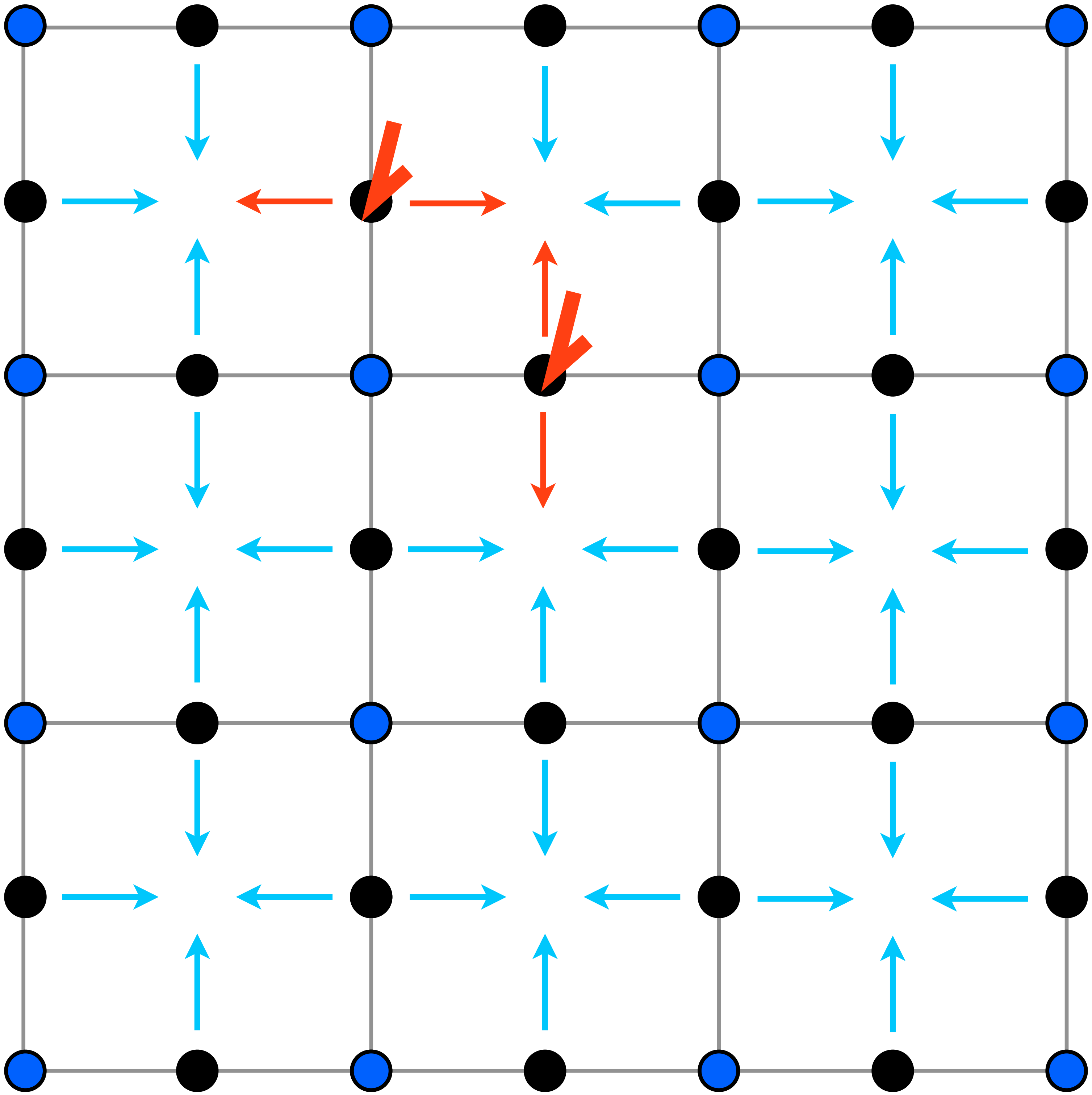


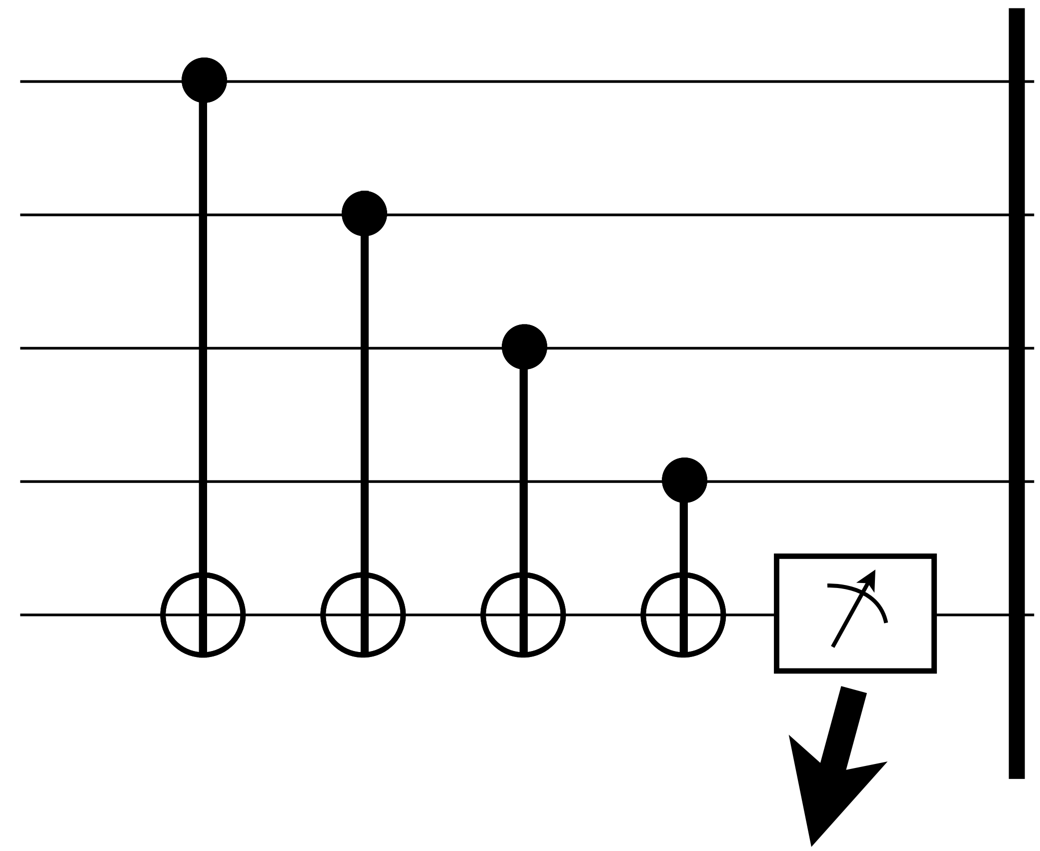
Memory Error 2





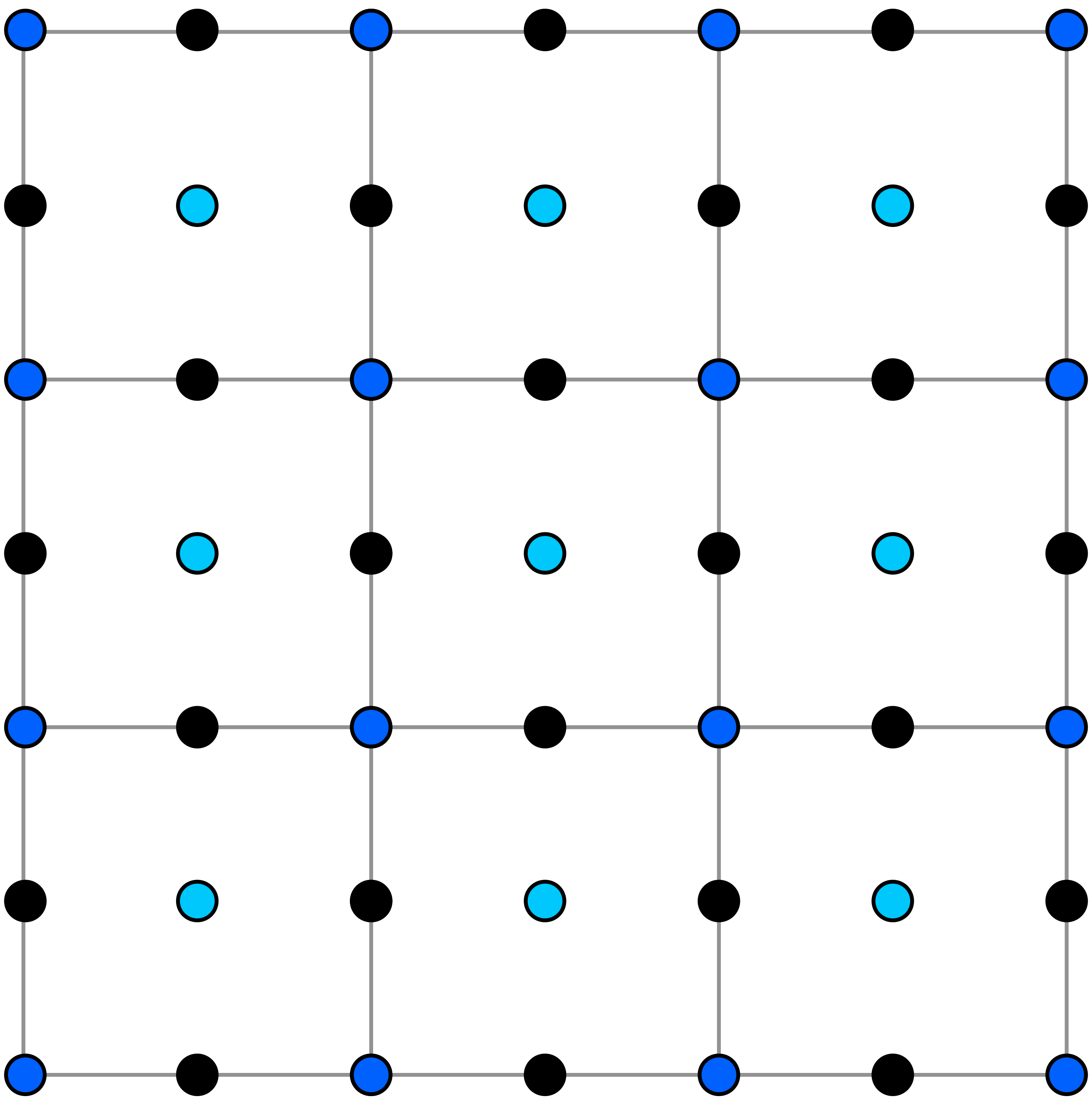
Memory Error 2

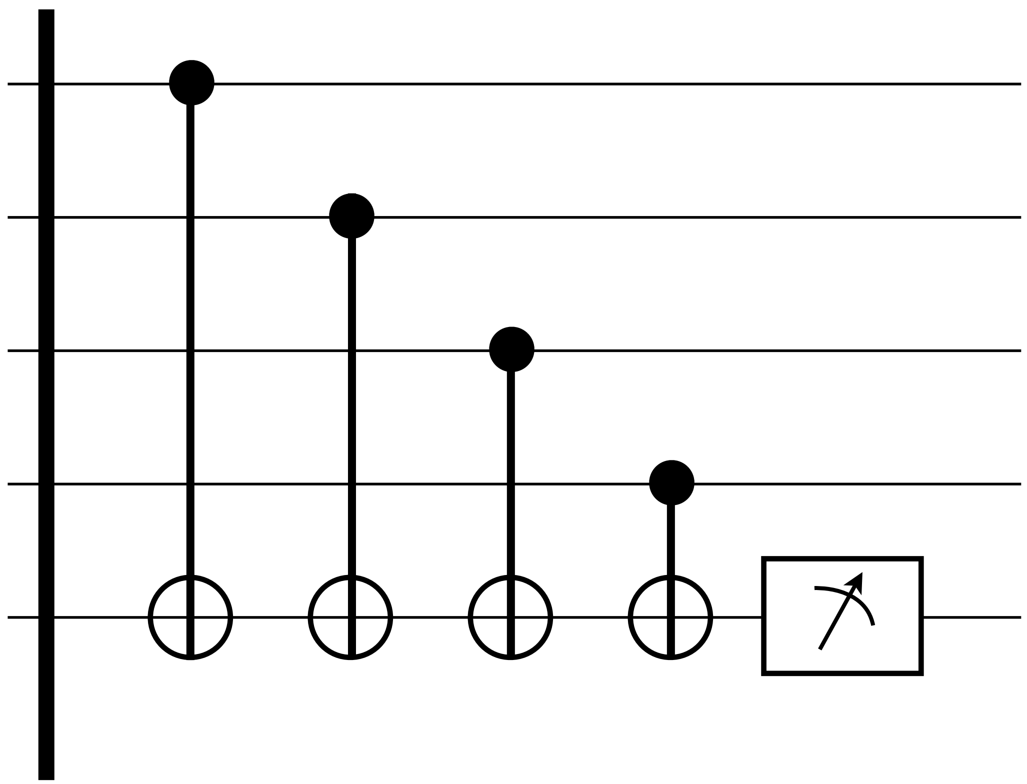




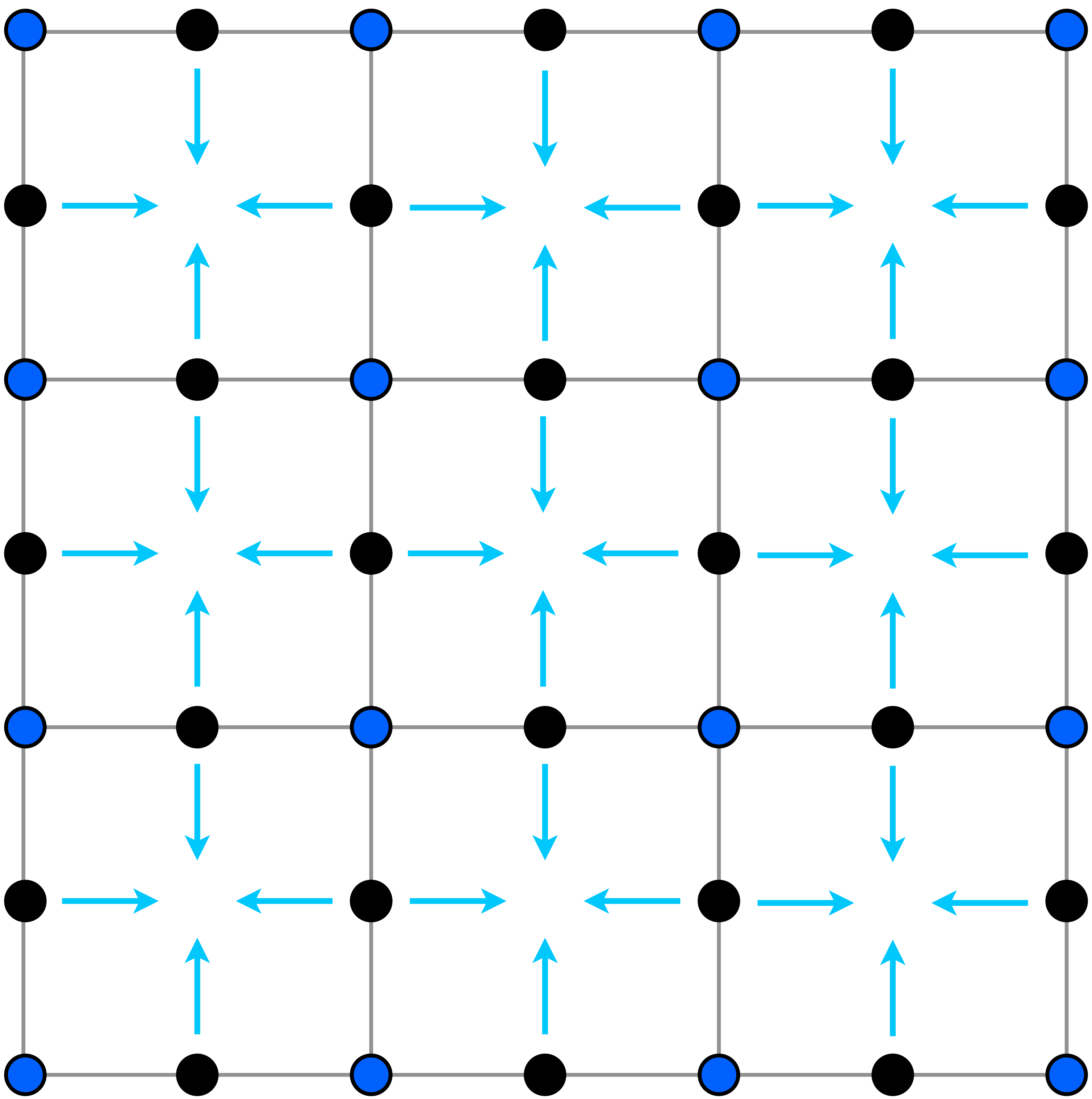
1	0	0
0	1	0
0	0	0

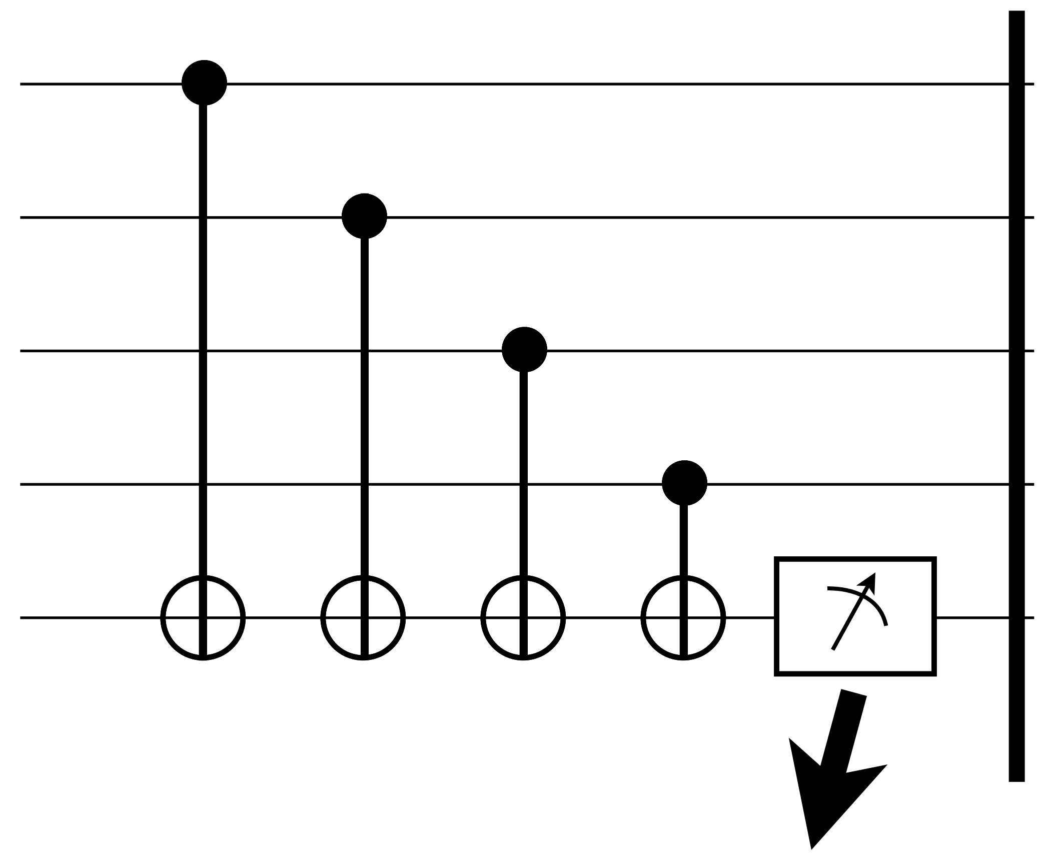
Measurement Error 1





Measurement Error 1

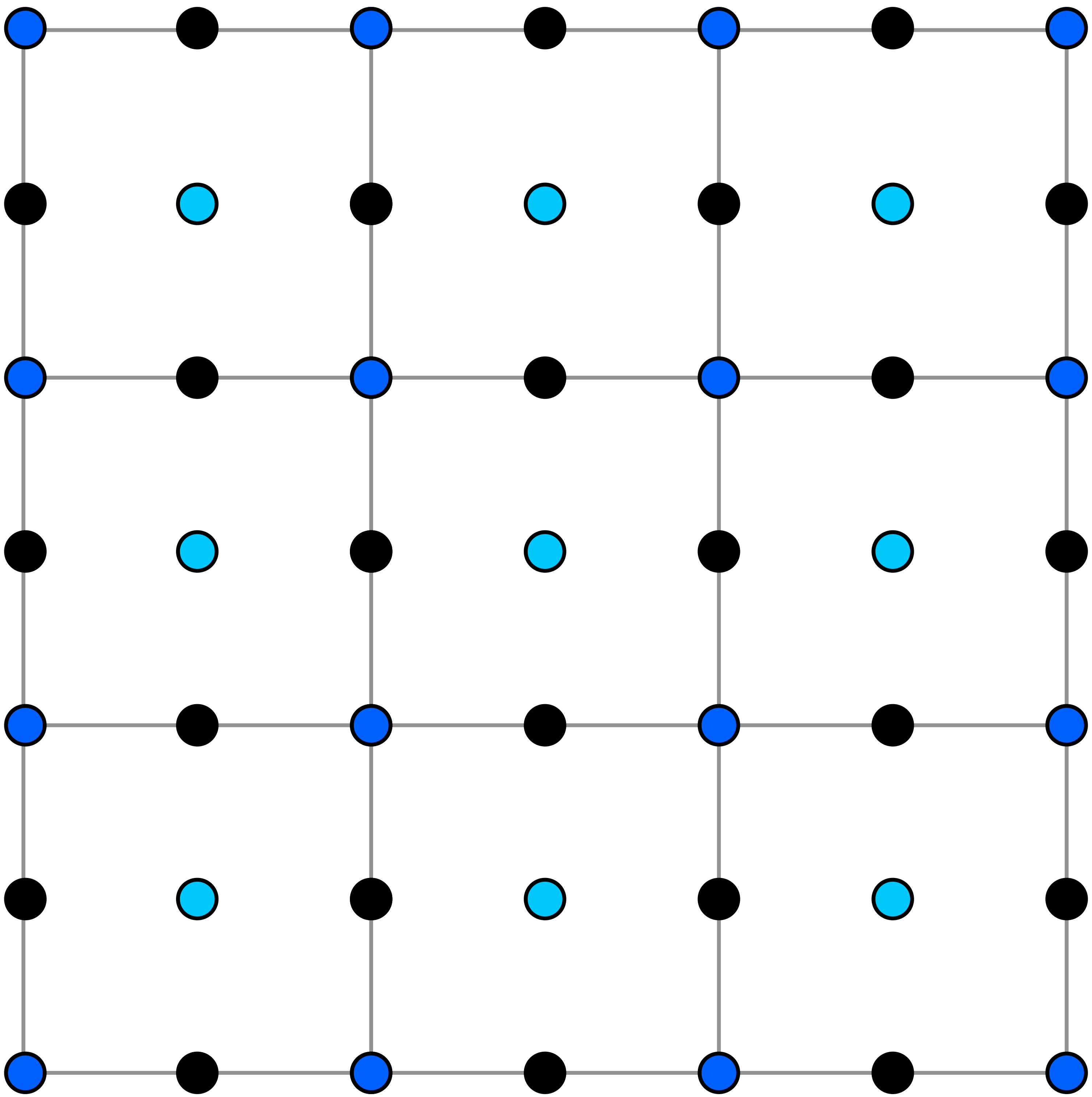


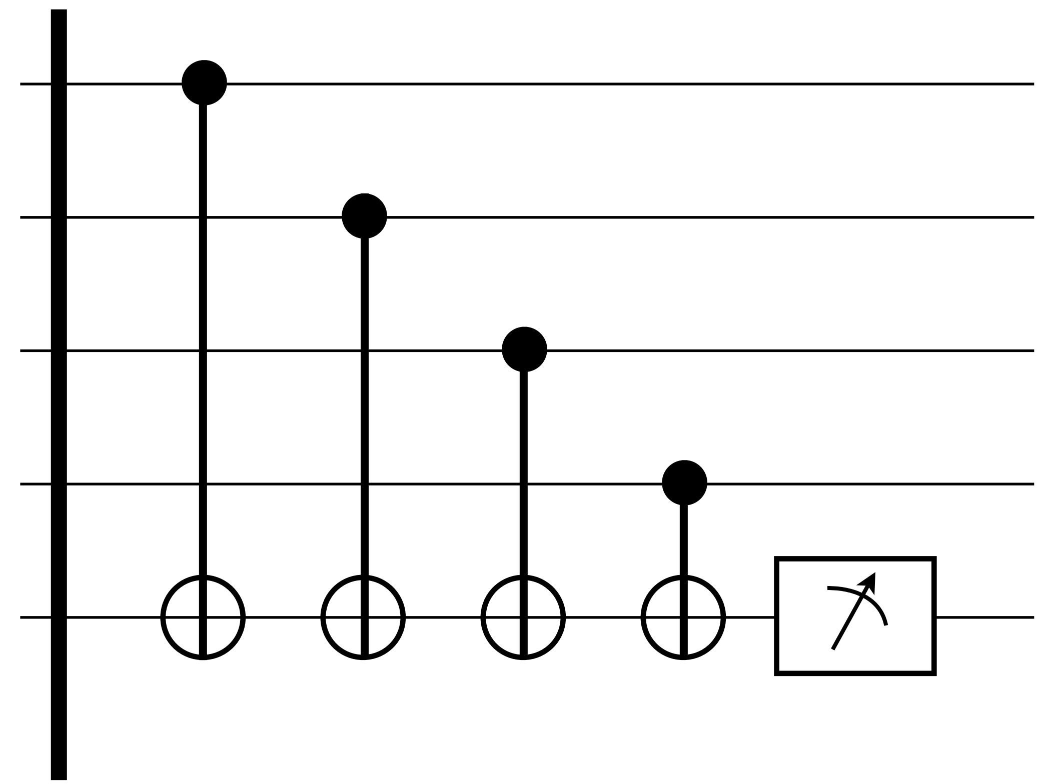


0	1	0
0	0	0
0	0	0

Measurement Error 2

Next Step

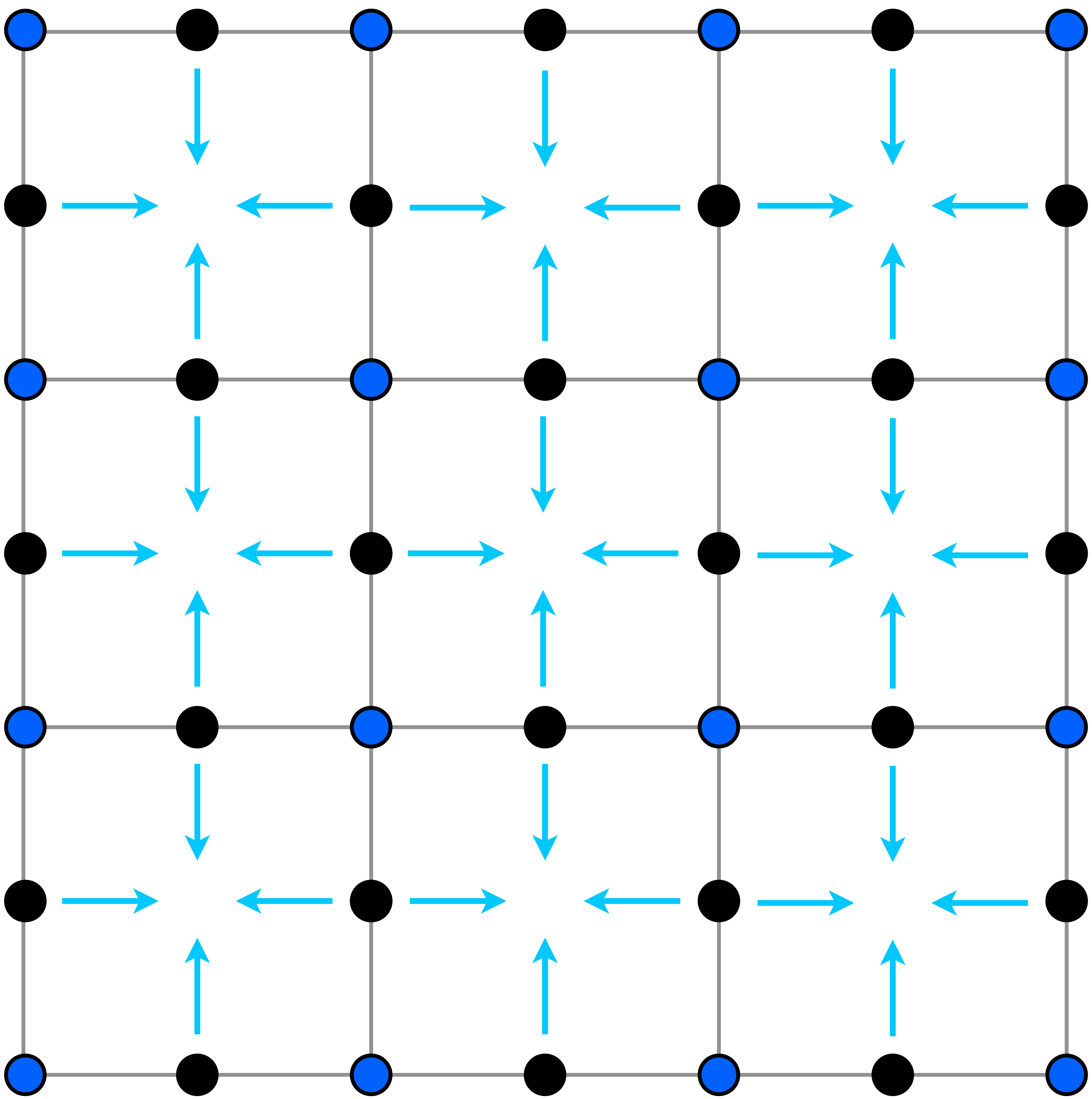


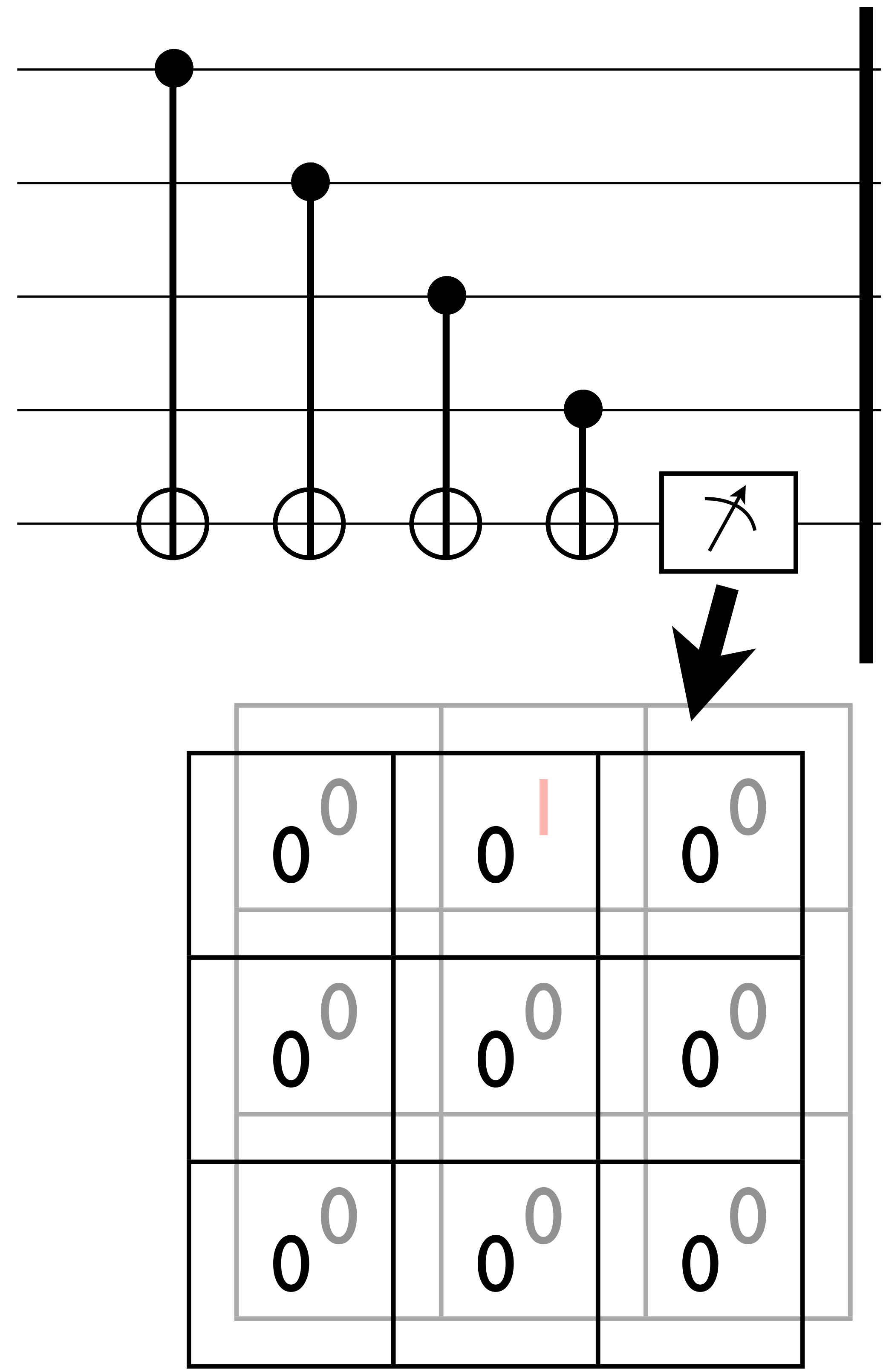


0	1	0
0	0	0
0	0	0

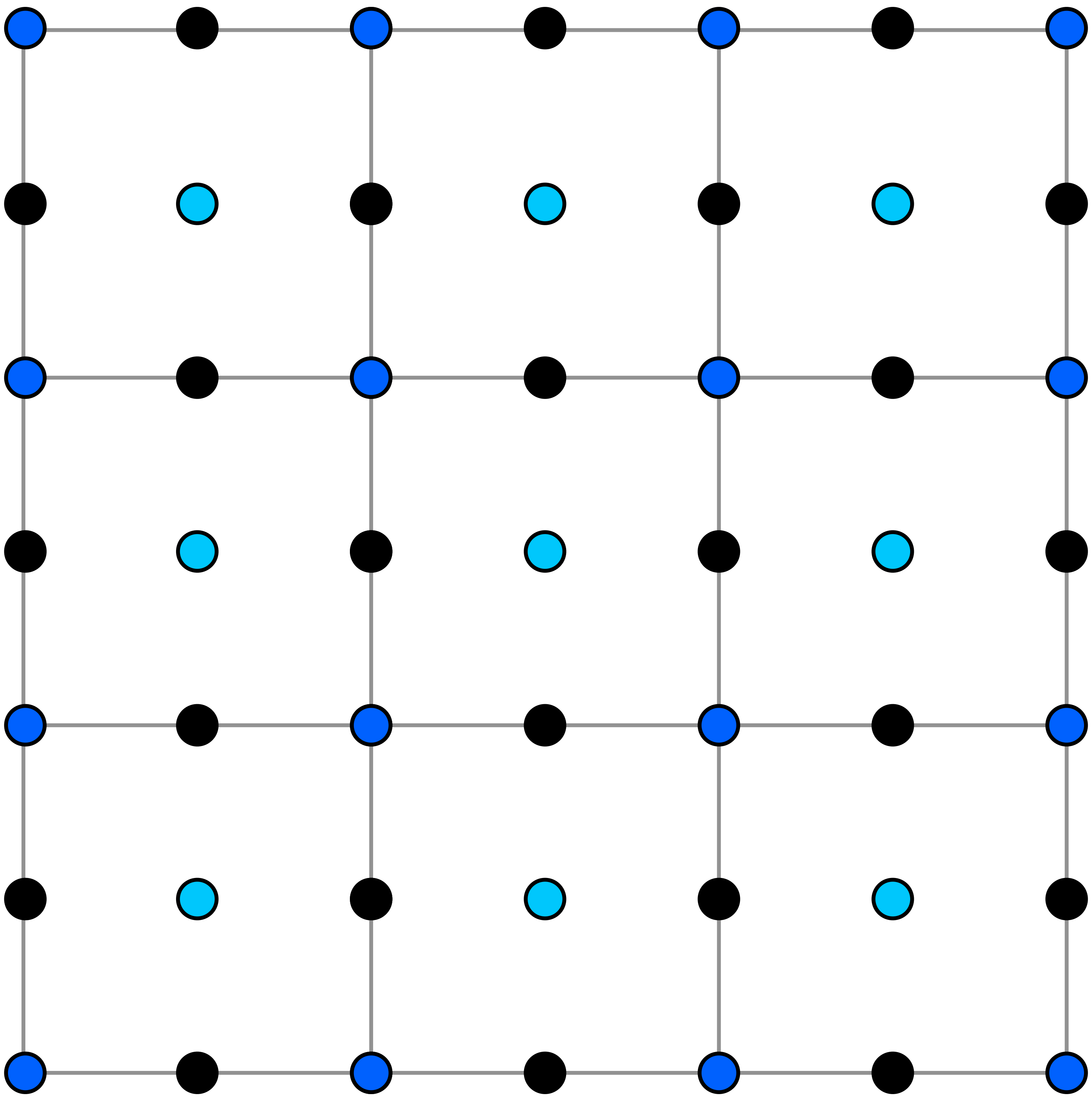
Measurement Error 2

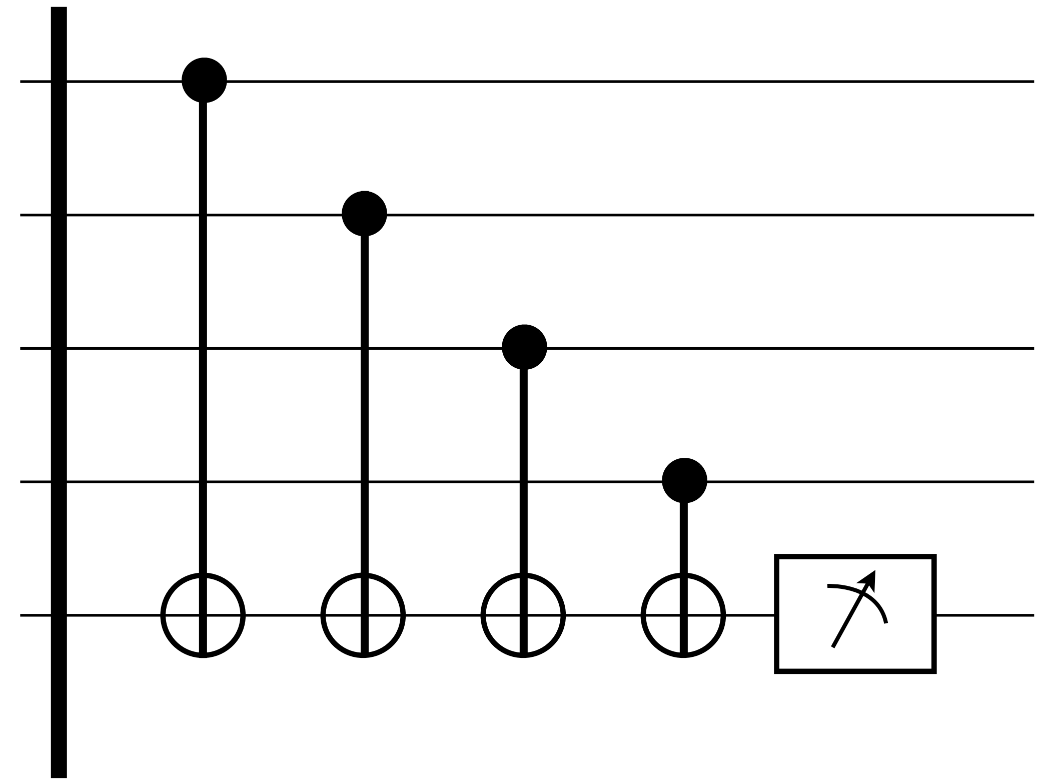
Next Step





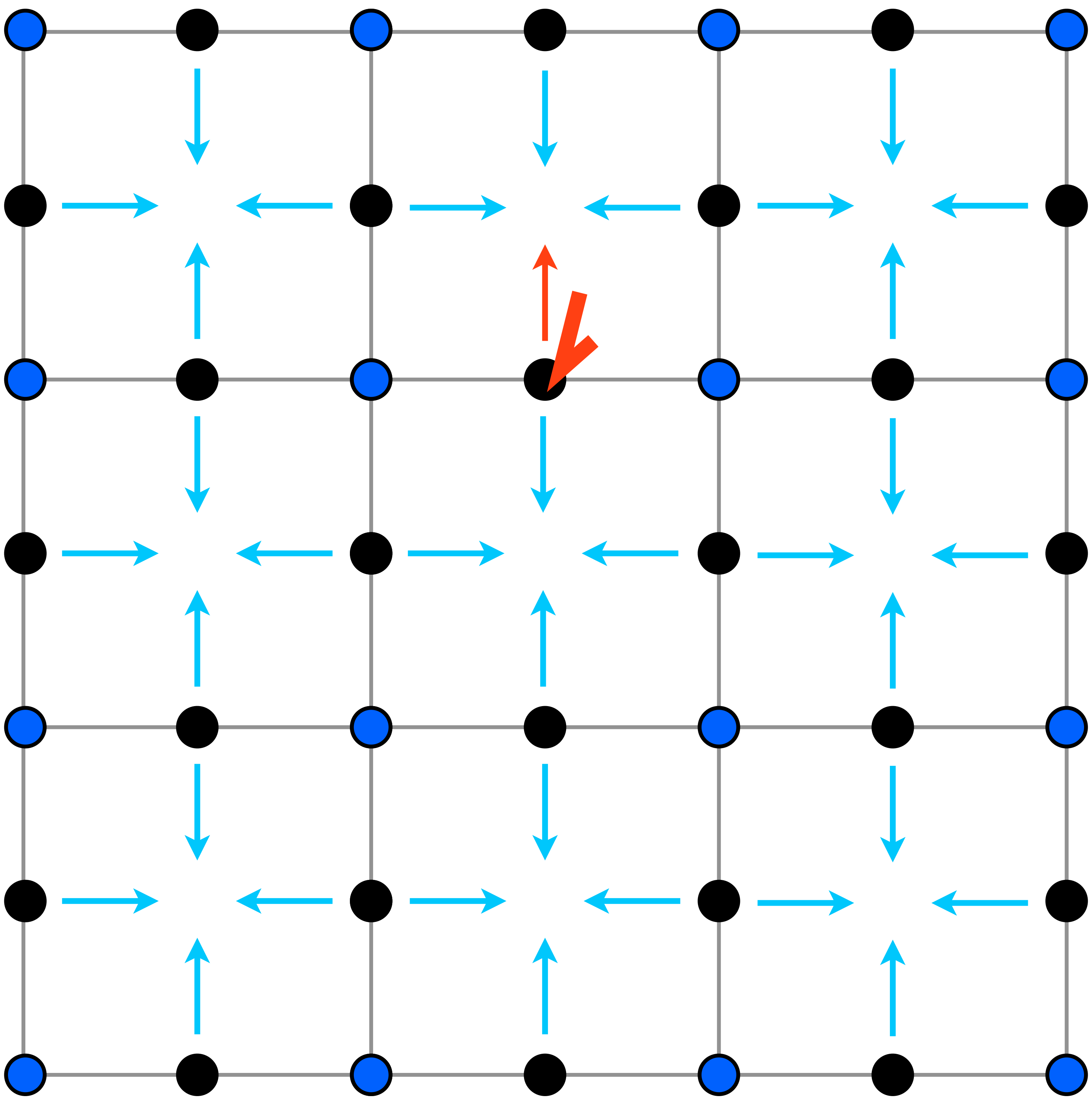
Gate Error 1

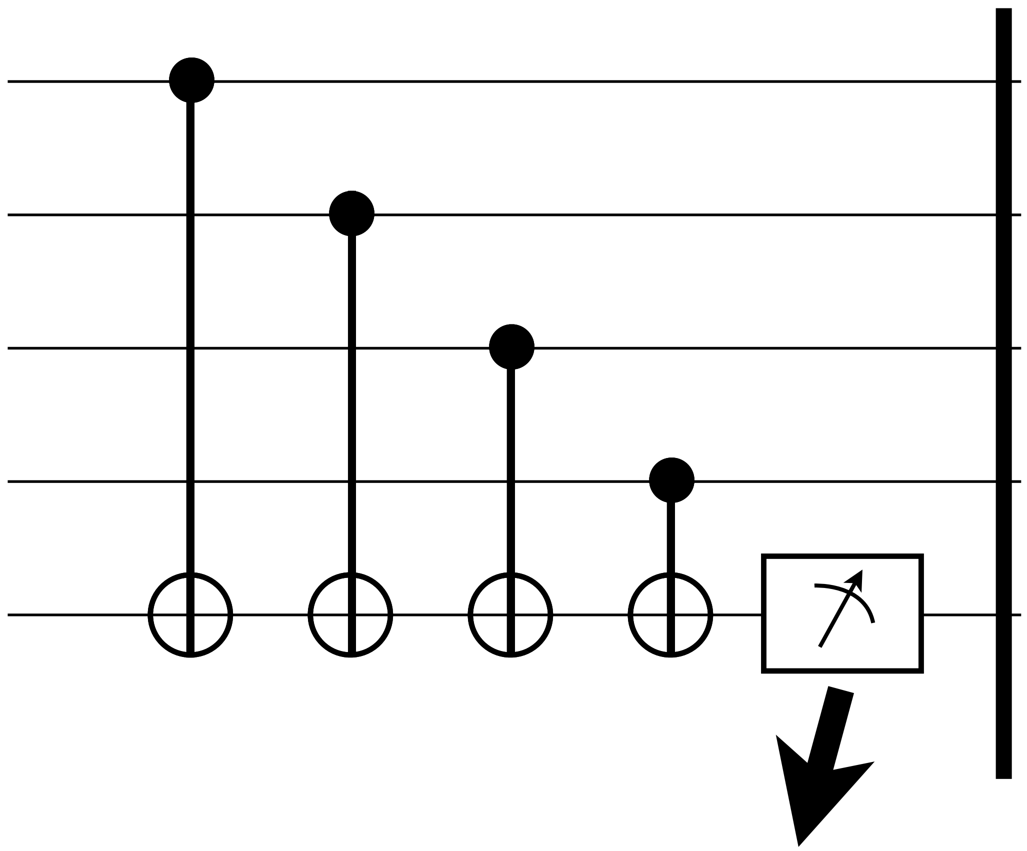




23

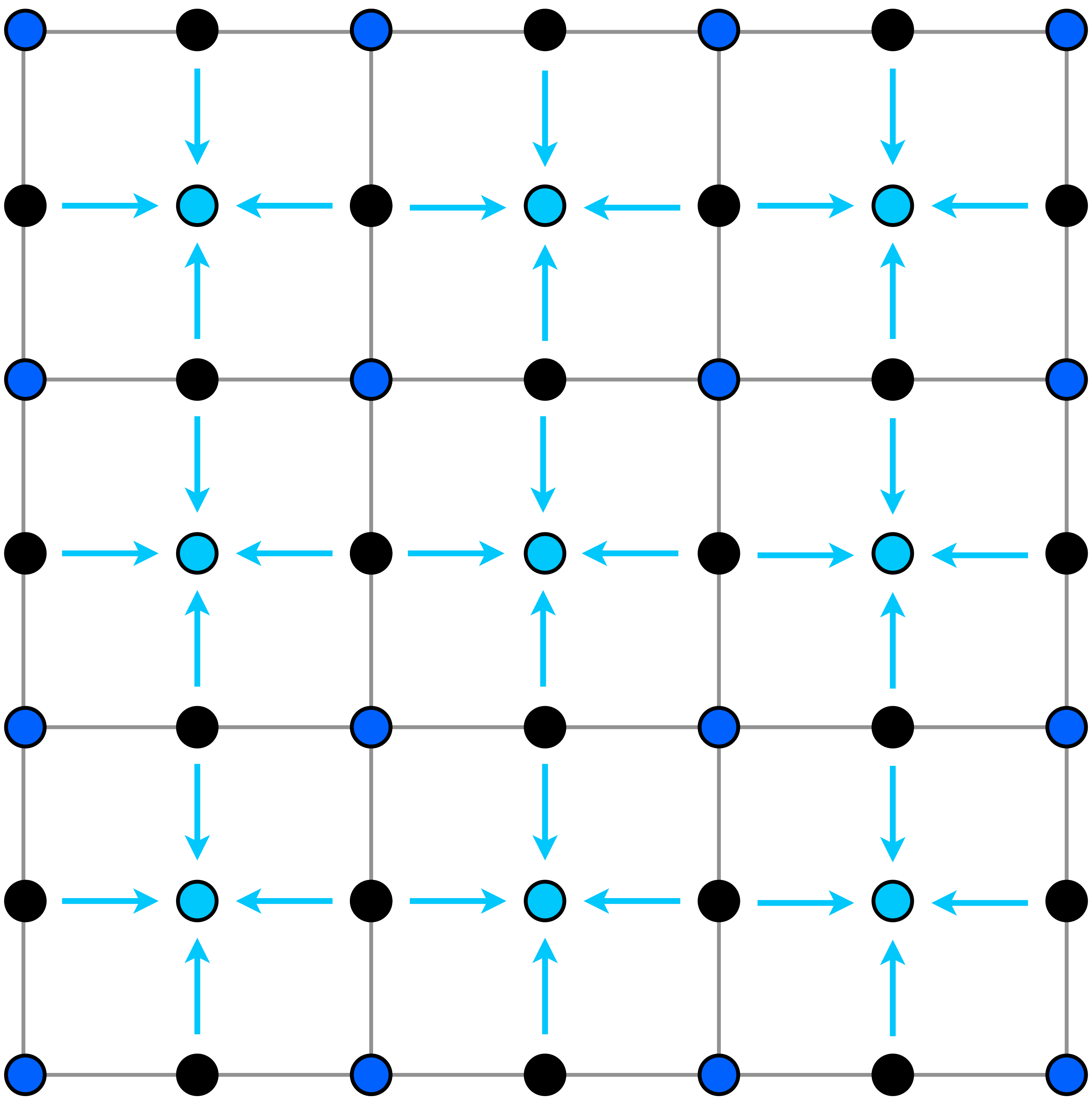
Gate Error 1

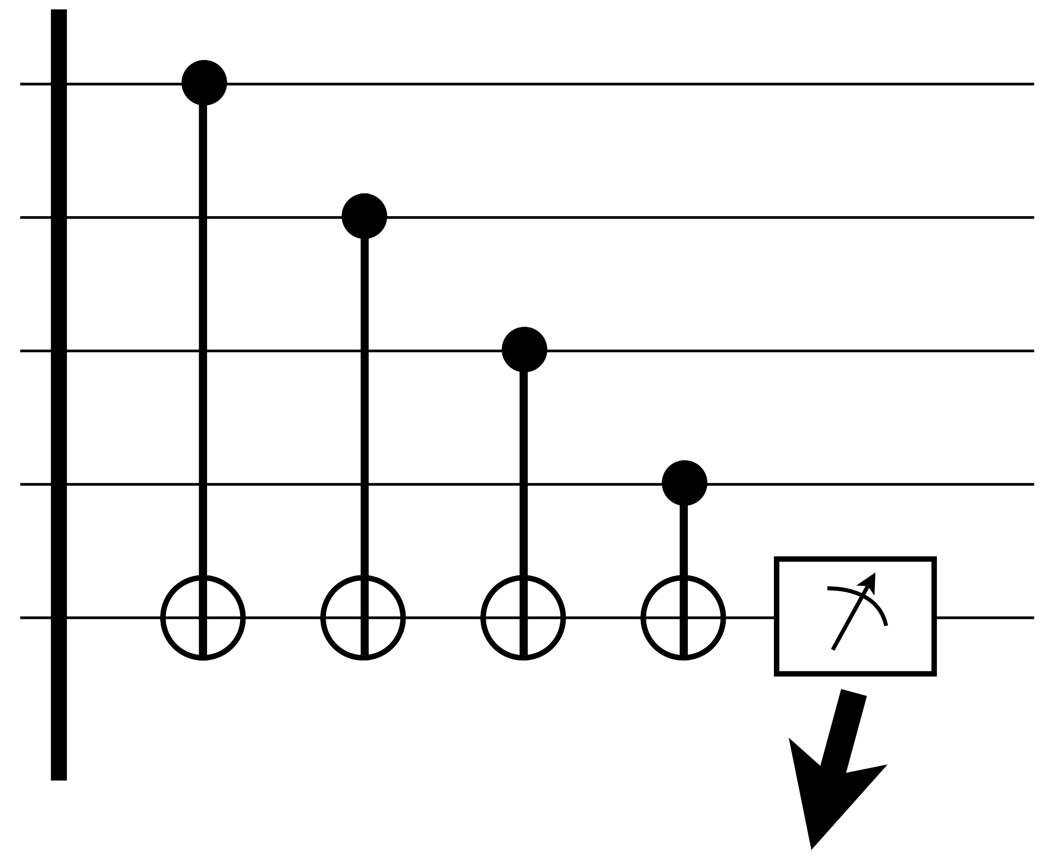




0	1	0
0	0	0
0	0	0

Measurement Error

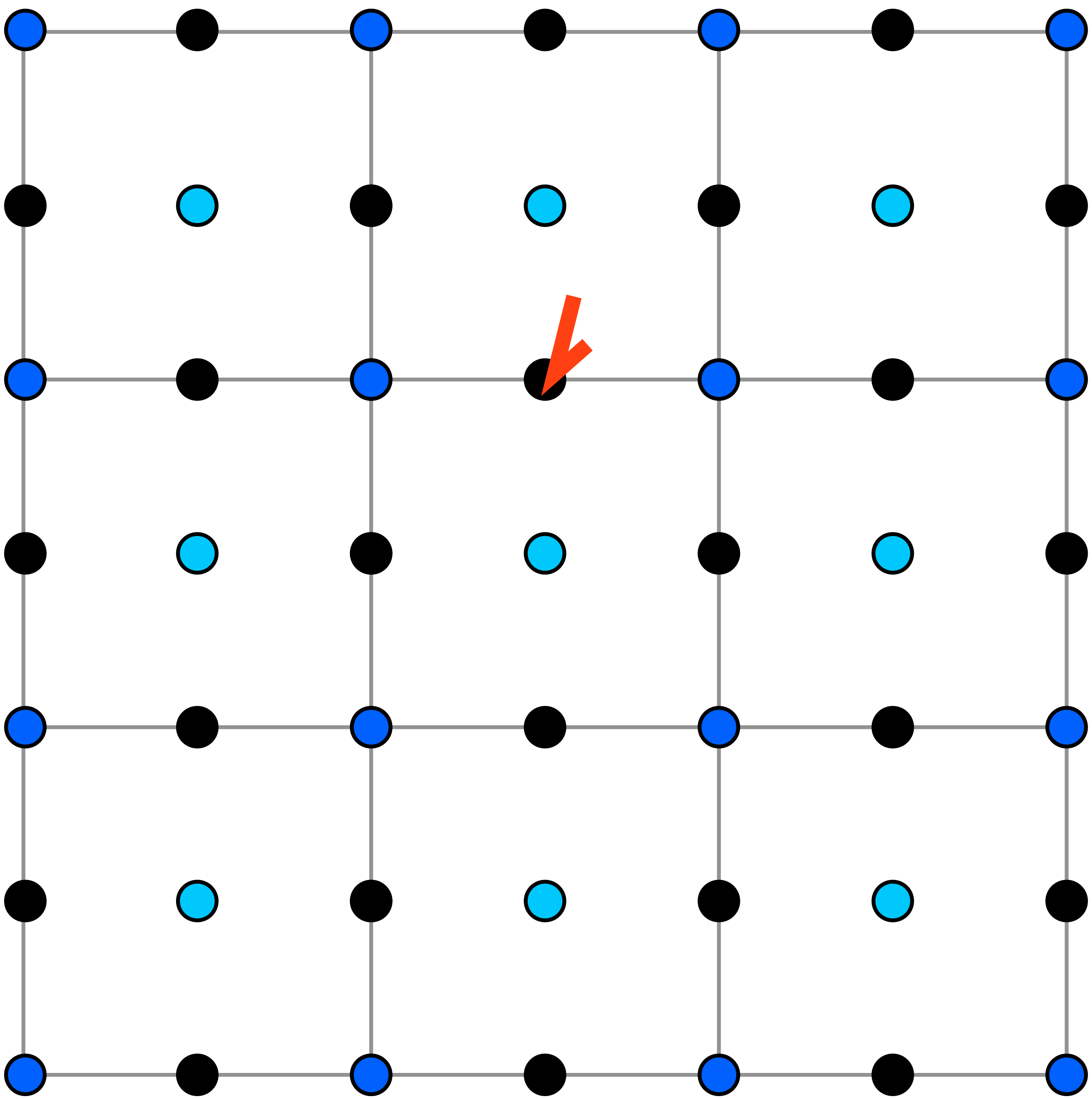


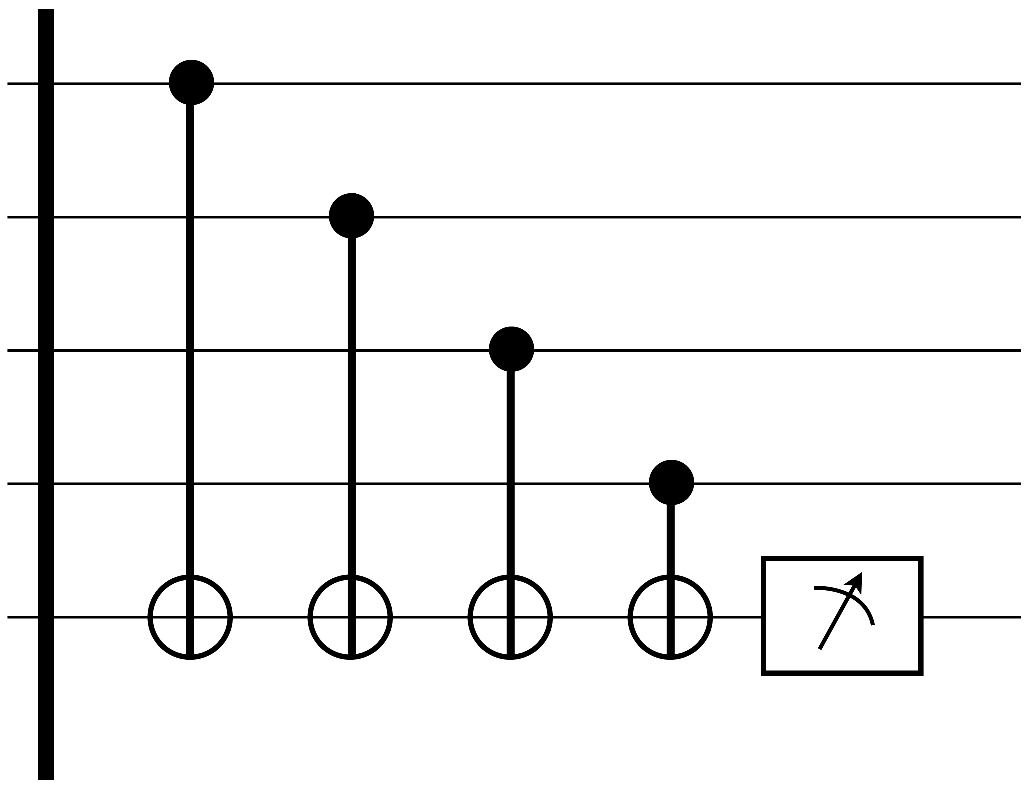


0	1	0
0	0	0
0	0	0

Gate Error 2

Next Step

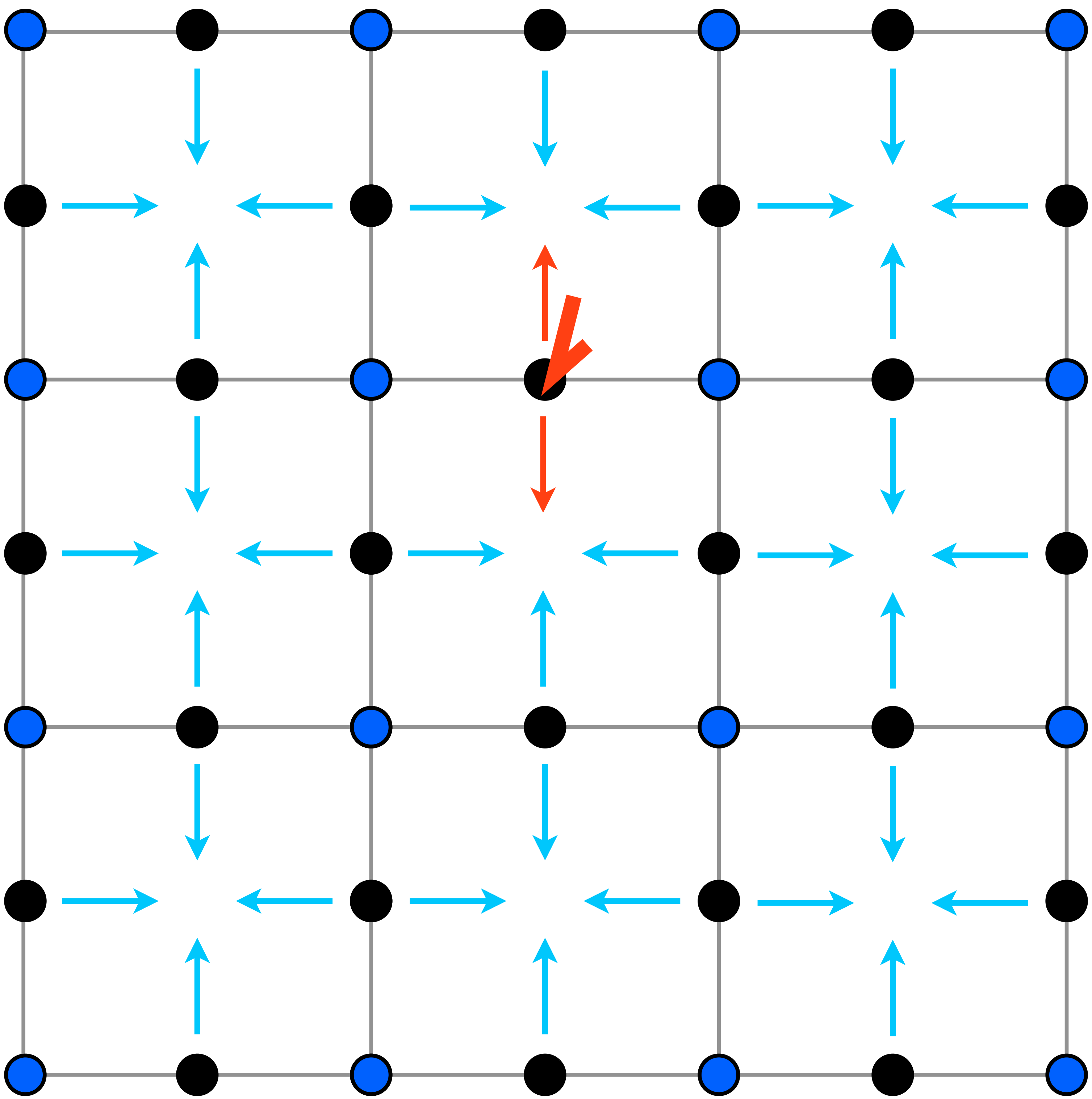


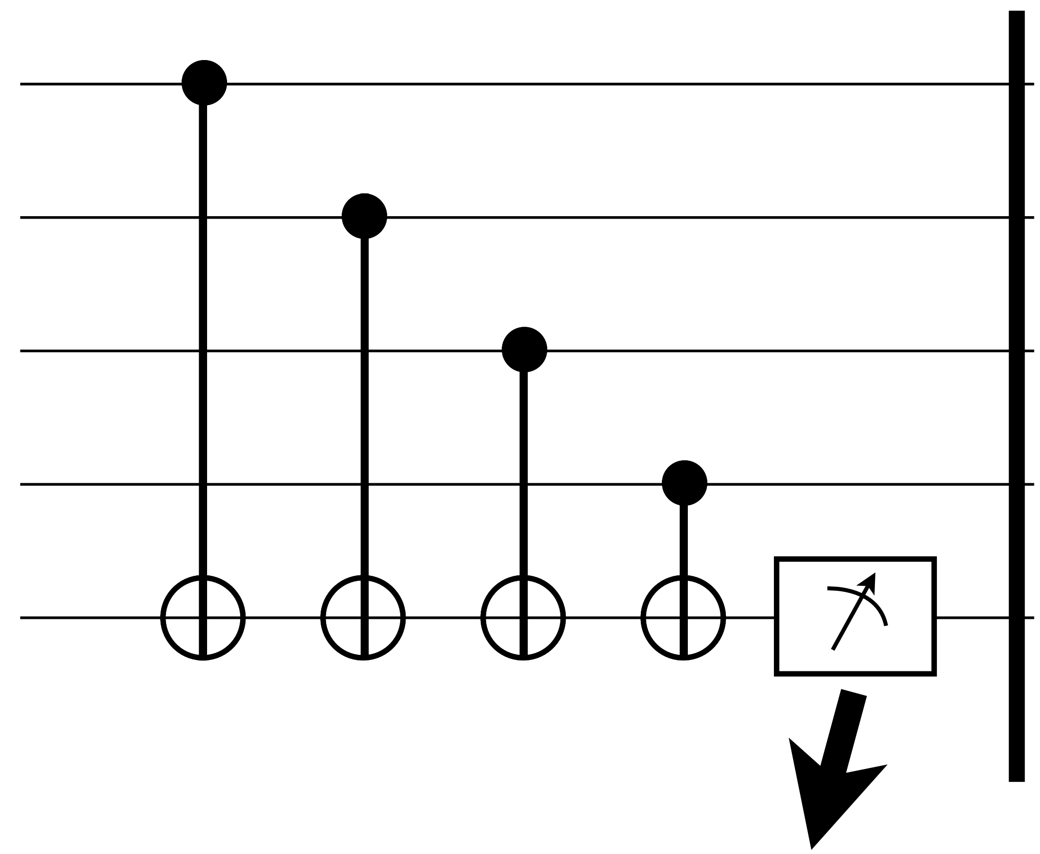


25

Gate Error 2

Next Step

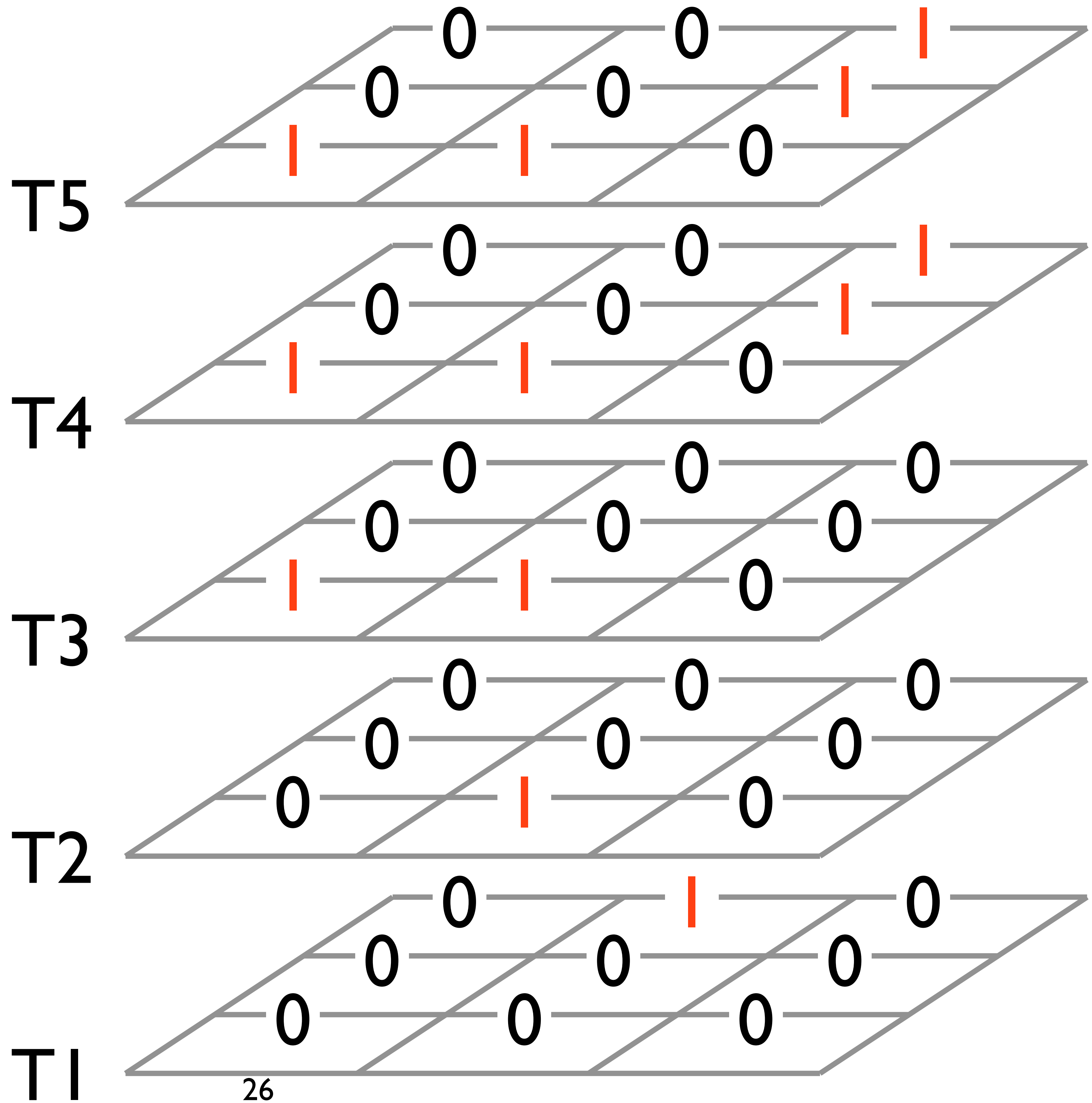




0	1	0
0	1	0
0	0	0

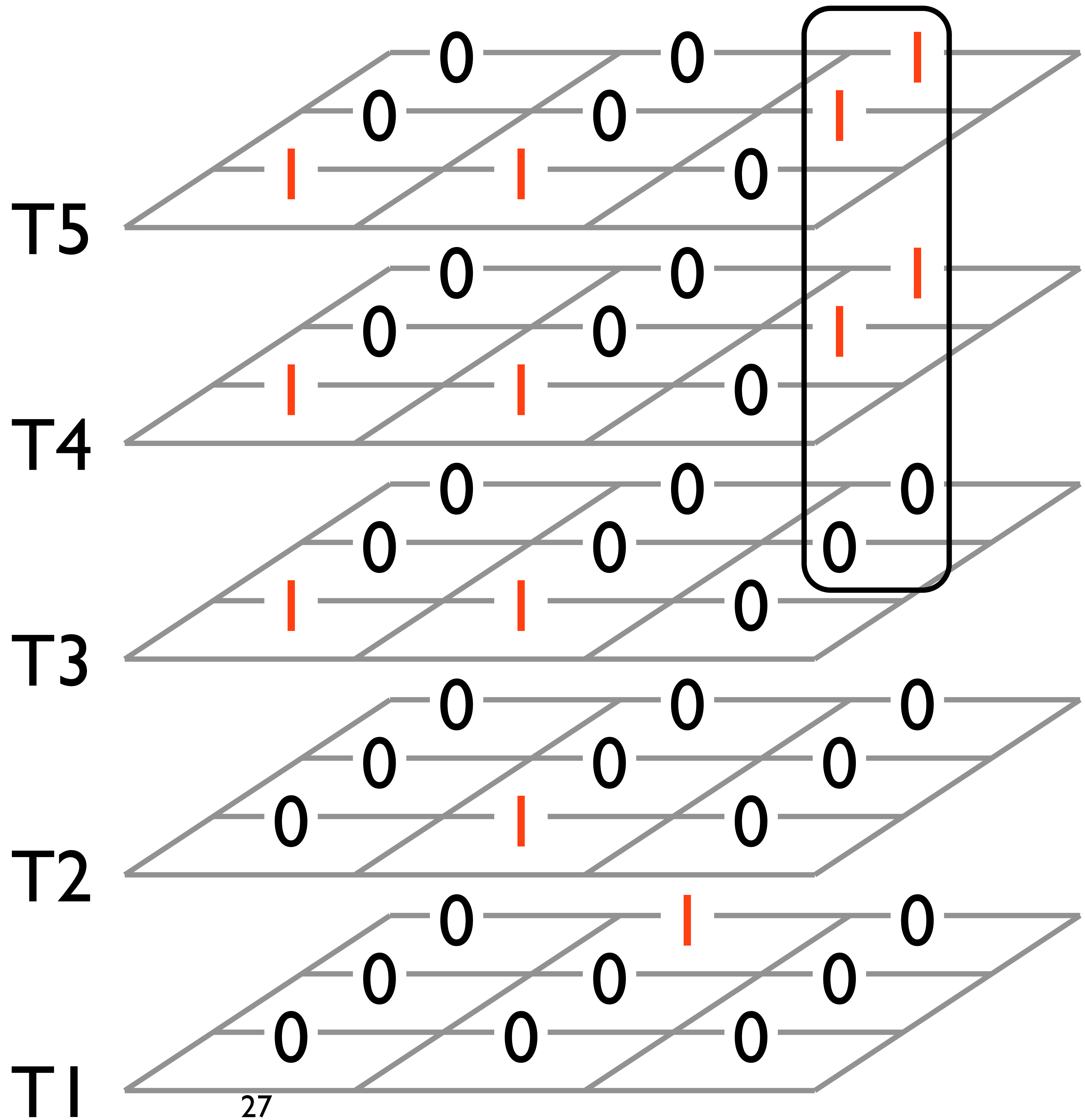
25

Error syndromes



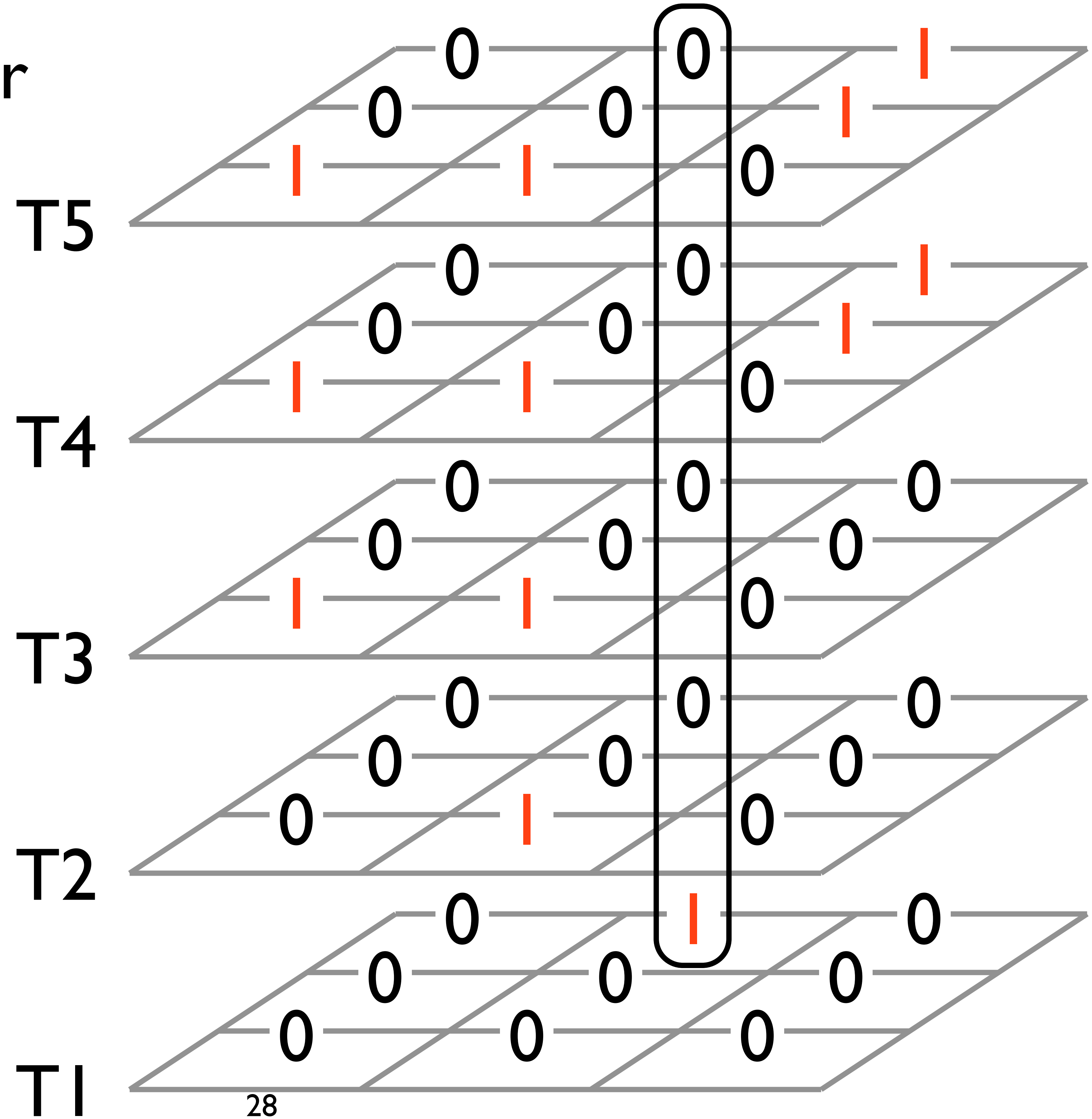
Error syndromes

memory error



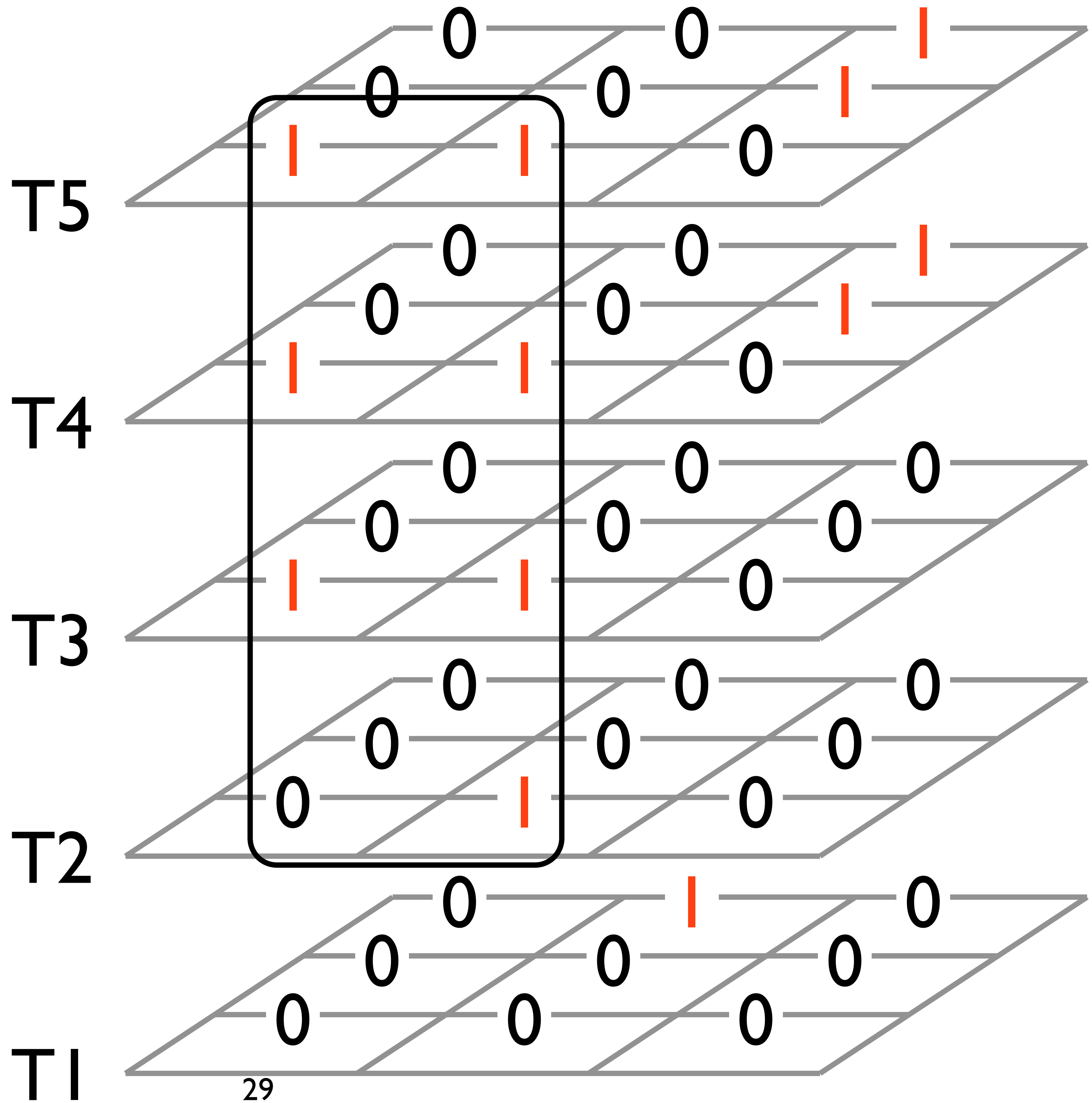
Error syndromes

measurement error



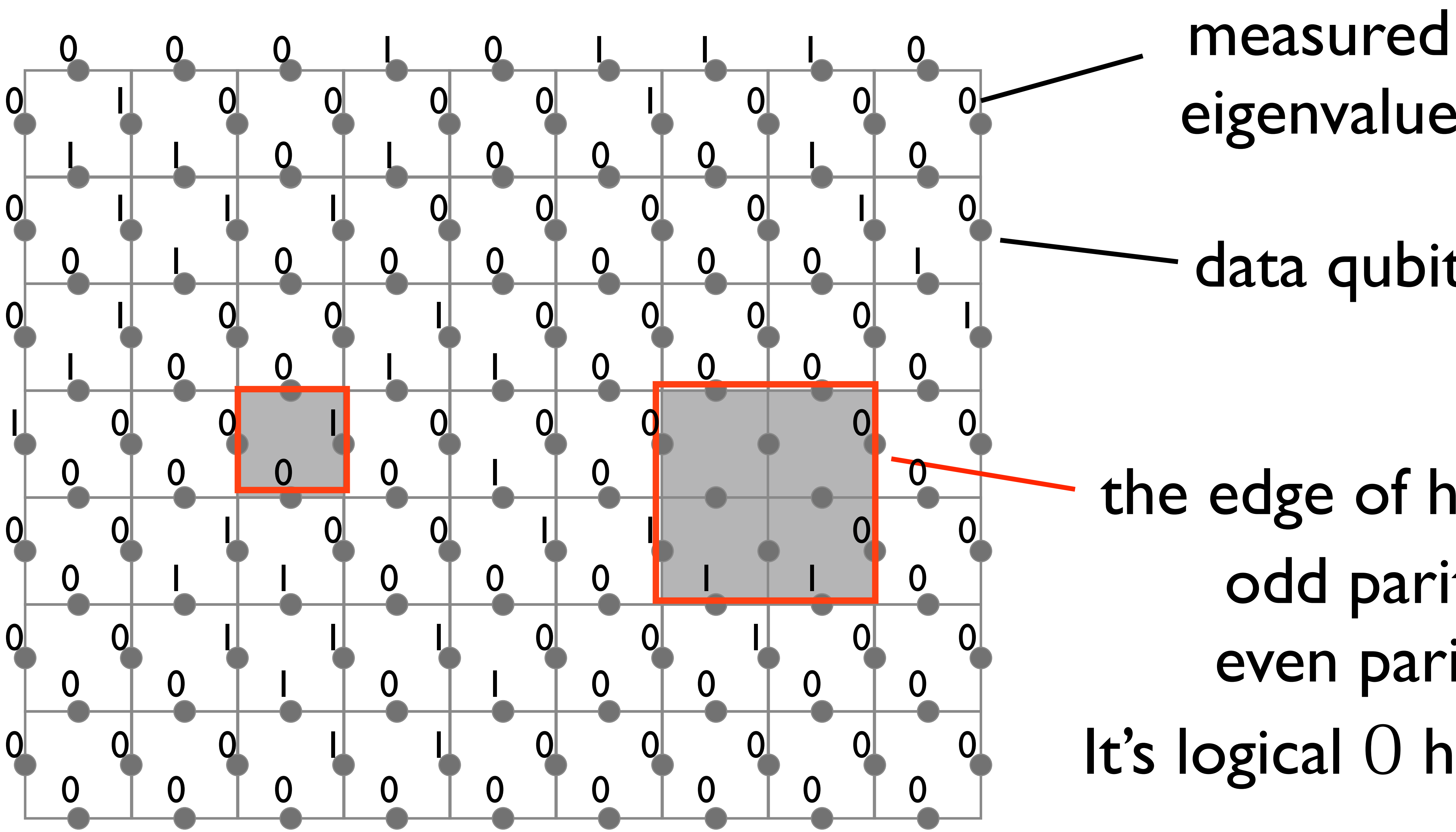
Error syndromes

gate error



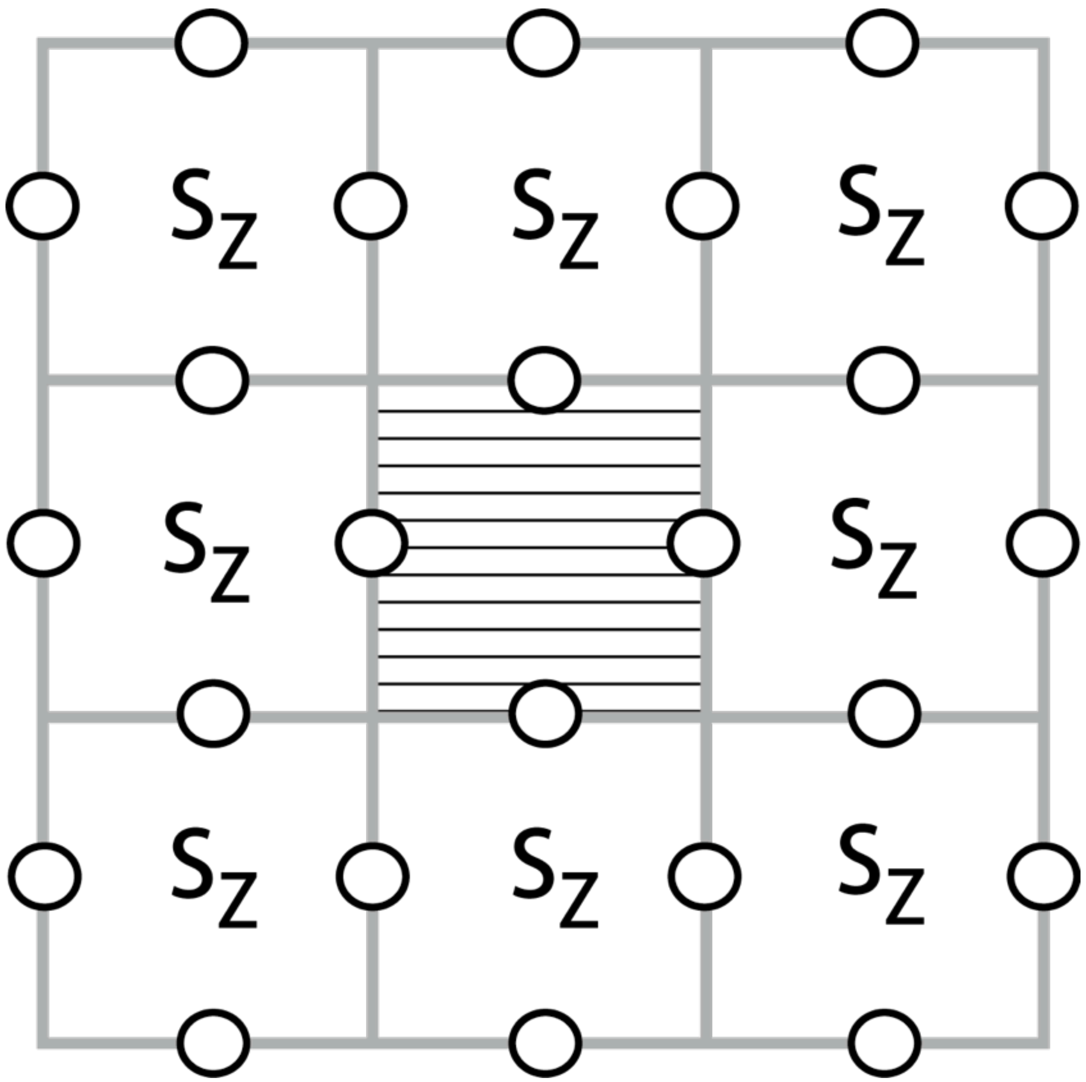
29

Simplest Example ~Encoding logical qubit~



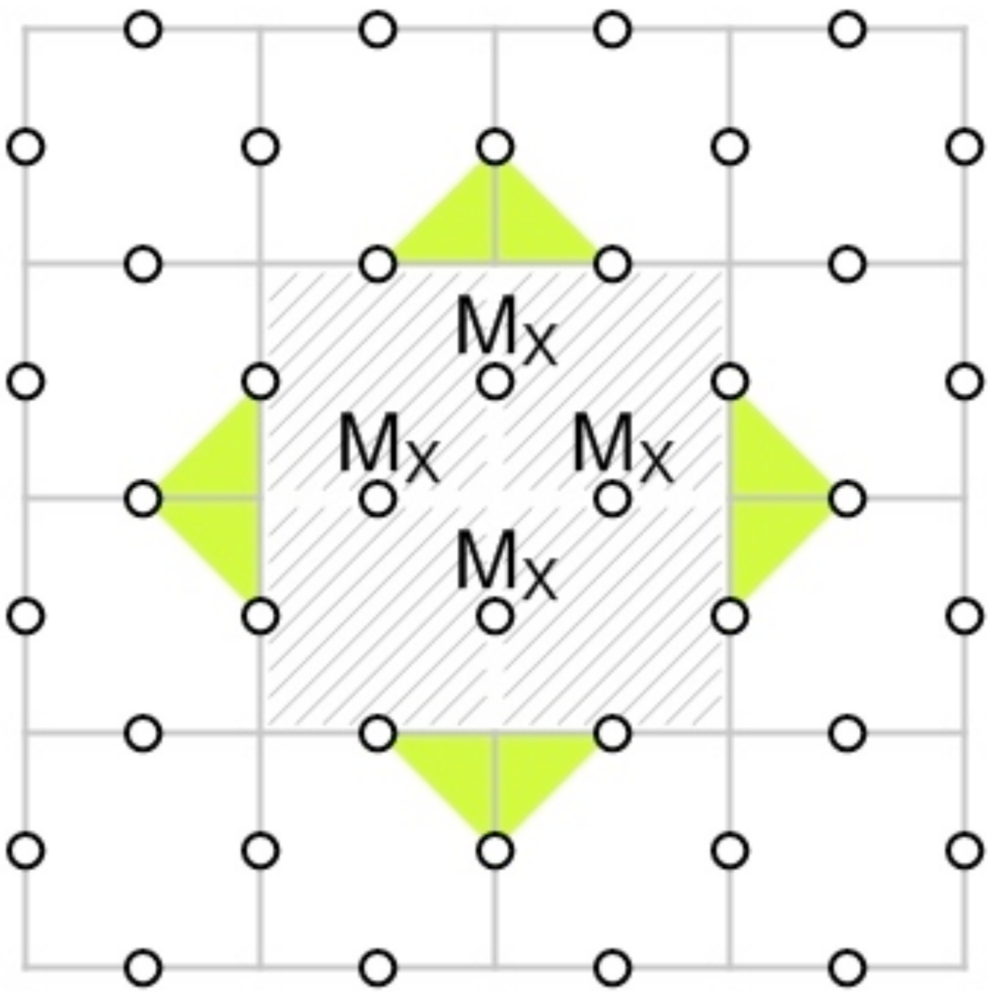
odd parity: 0
even parity: 1
It's logical 0 here.

Lattice holes as logical qubits



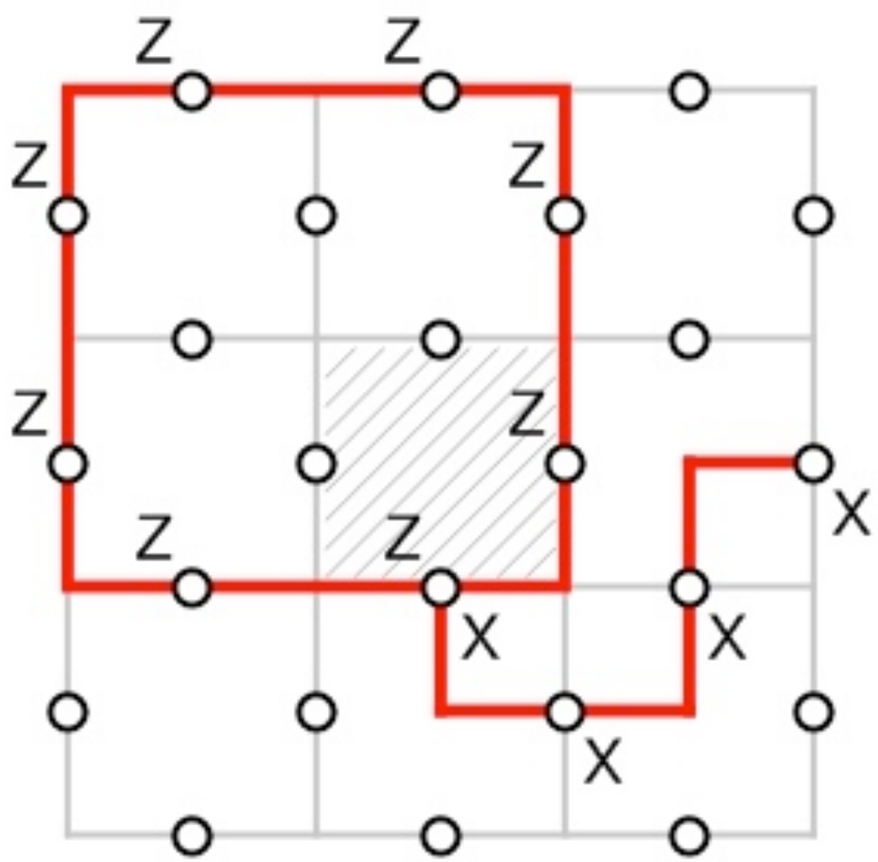
- Introduce degree of freedom by not measuring stabilizer (first, measure out any data qubits in the *interior* of the hole in X basis, to disentangle)
- Only one degree of freedom associated with arbitrary size hole
- Here, 24 lattice qubits, 9 Z stabilizers, 16 X stabilizers (but one *not independent*) = 24 independent stabilizers
- n.b.: Holes referred to as "defects" in most papers, I reserve that term for physically defective qubits

Larger holes

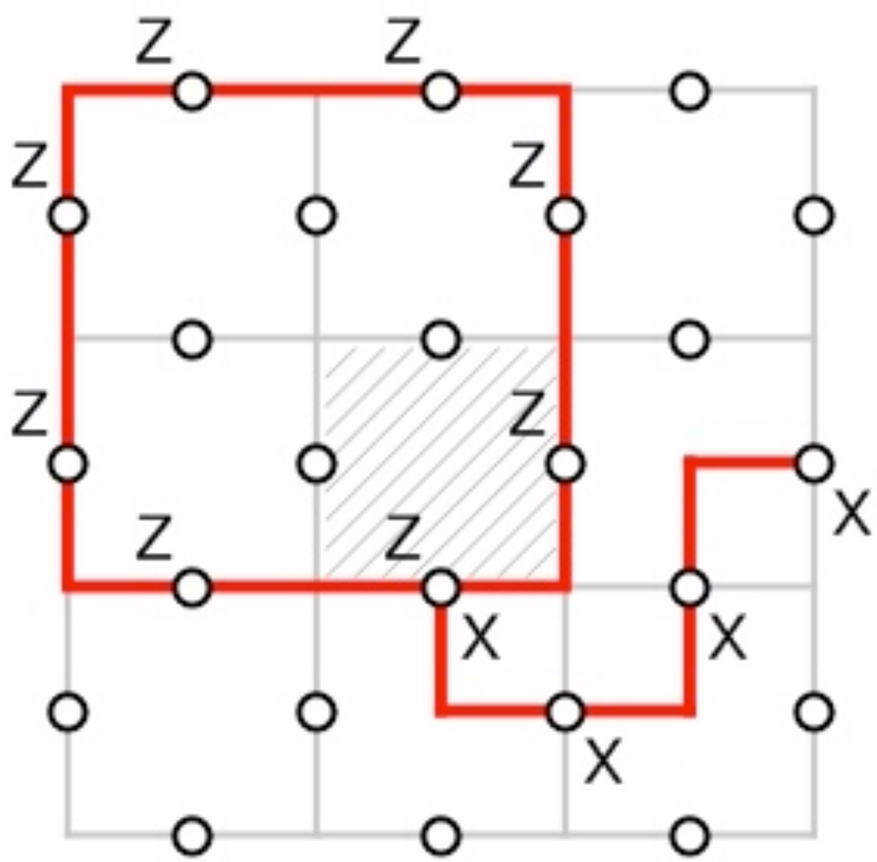


- Only one degree of freedom associated with arbitrary size hole
- Must correct three-term X stabilizers (green)
- Initialize to $|0_L\rangle$

Logical X and Z gates



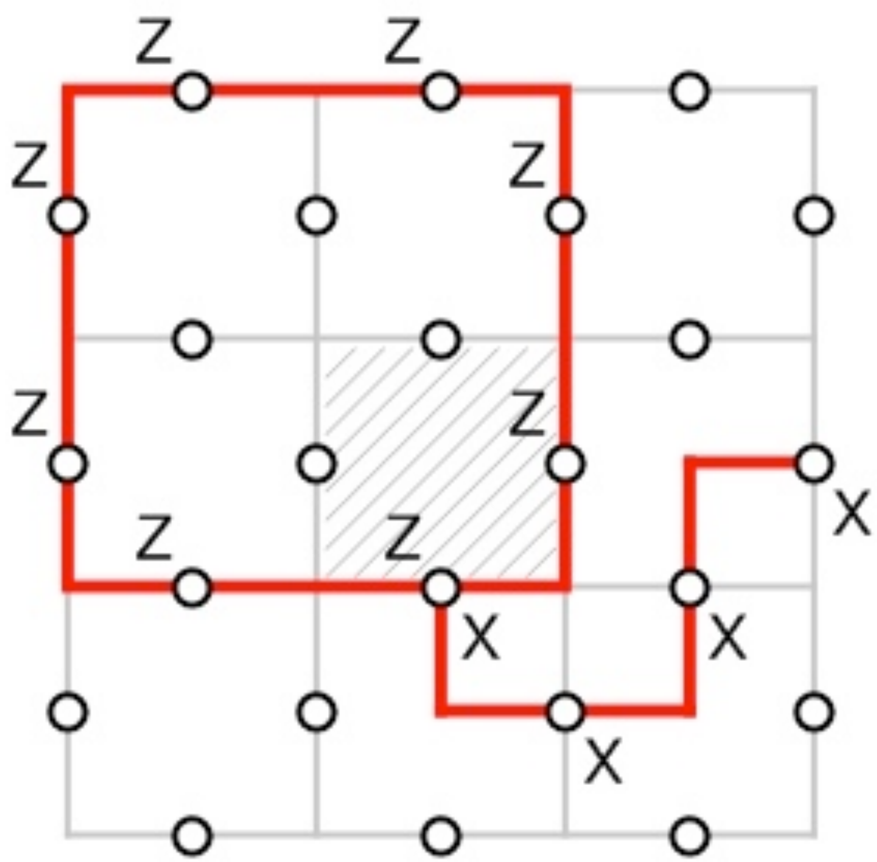
Logical X and Z gates



- Degree of freedom can be operated on by chains and rings of single-qubit X and Z *on lattice qubits*

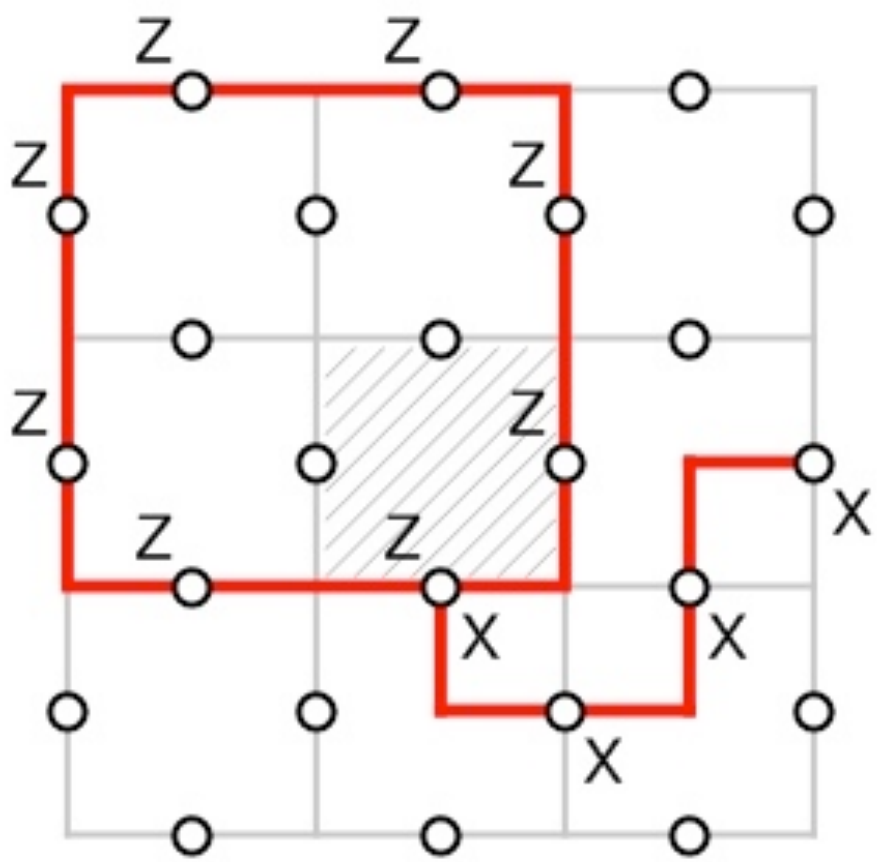
33

Logical X and Z gates



- Degree of freedom can be operated on by chains and rings of single-qubit X and Z *on lattice qubits*
- Chain of X flips connecting hole to boundary gives X gate on logical qubit

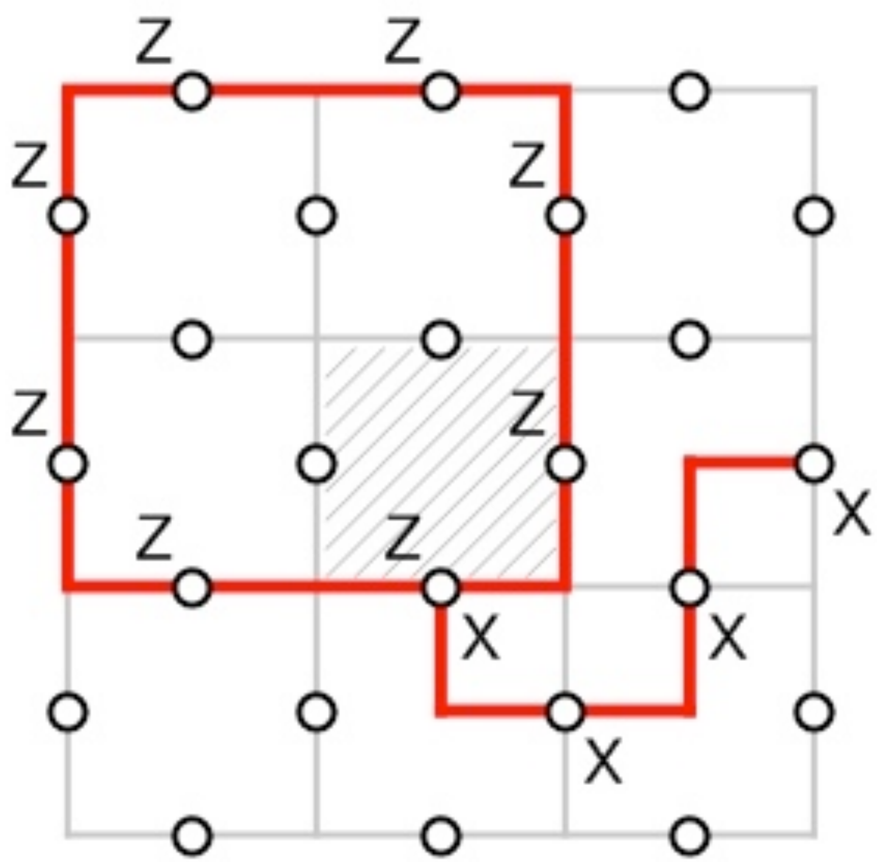
Logical X and Z gates



- Degree of freedom can be operated on by chains and rings of single-qubit X and Z *on lattice qubits*
- Chain of X flips connecting hole to boundary gives X gate on logical qubit
- Ring of Z flips *around* hole gives logical Z gate

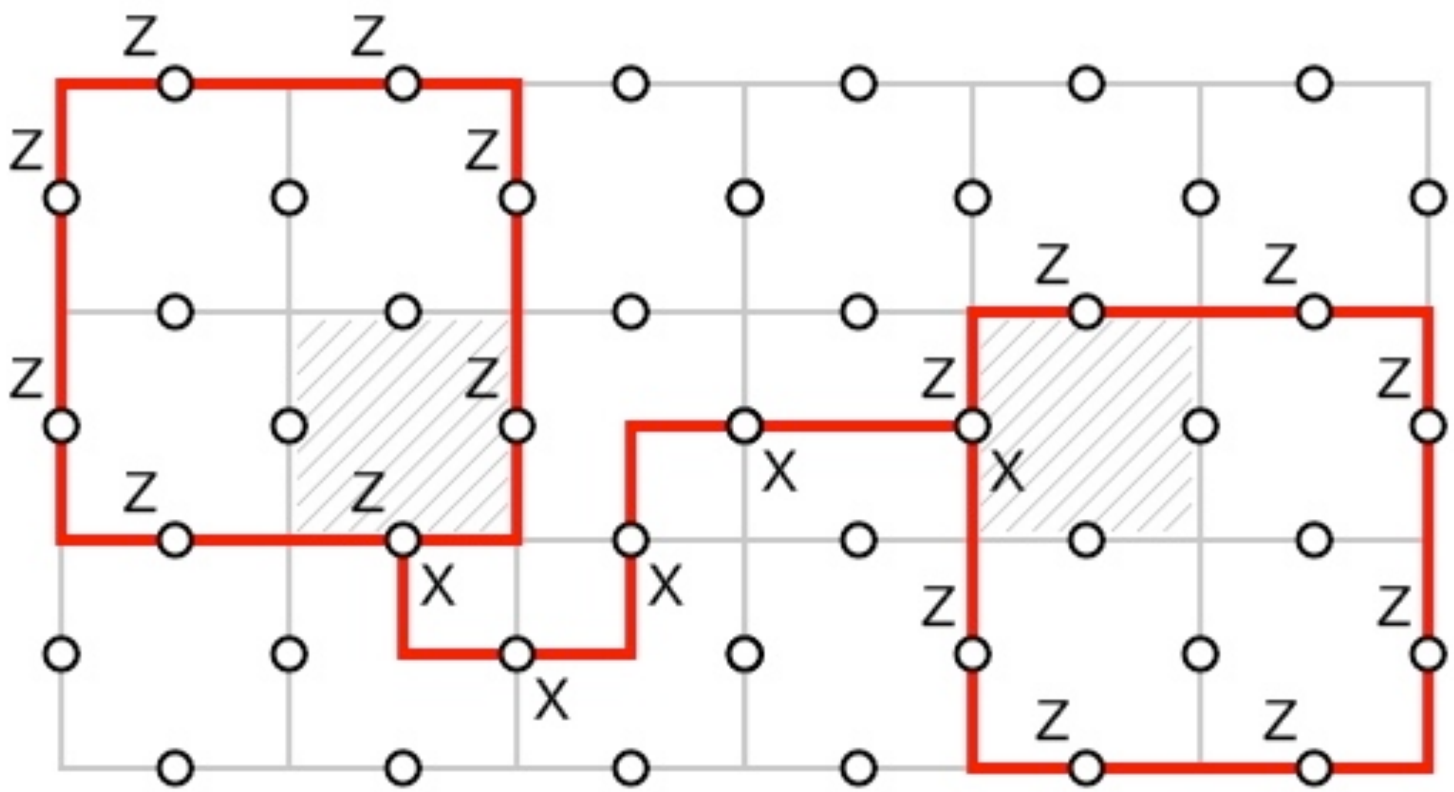
Aqua : Advancing Quantum Architecture

Logical X and Z gates



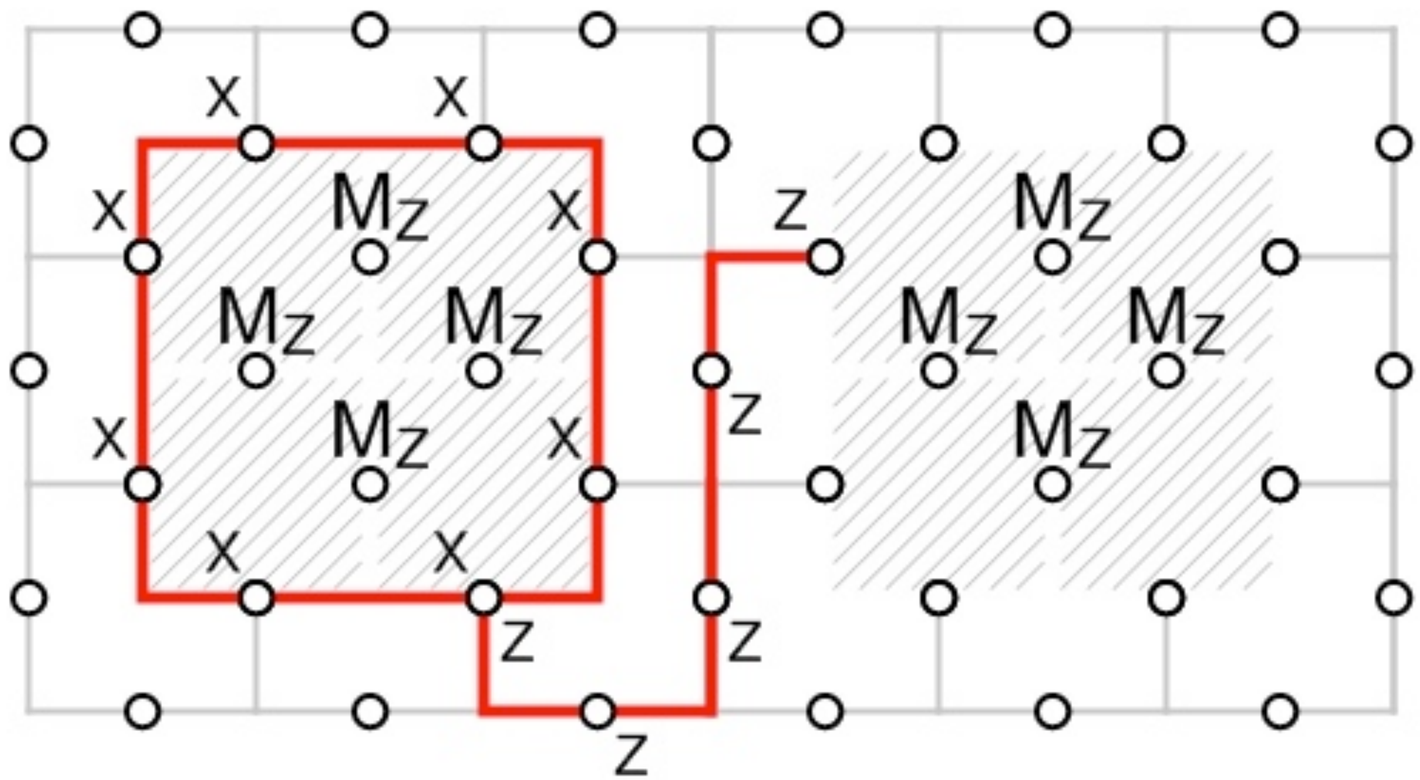
- Degree of freedom can be operated on by chains and rings of single-qubit X and Z *on lattice qubits*
- Chain of X flips connecting hole to boundary gives X gate on logical qubit
- Ring of Z flips *around* hole gives logical Z gate
- n.b.: This can be *any* ring around the hole or *any* chain connecting hole to lattice edge

Pairing up holes



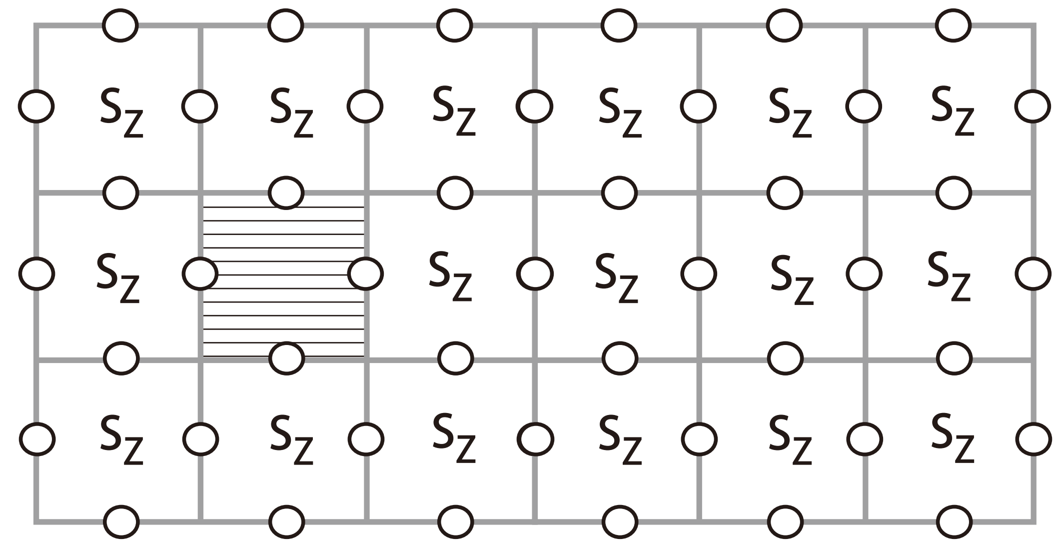
- Better to use a pair of defects to represent one qubit
  - No need to connect operators to boundary
  - Easier to move qubit around computer
- Independent adjustment of X/Z error correction strength
- Terminology: **smooth qubit**
- Logical X gate chain connects the holes
- Logical Z gate circles *one* of the holes

Rough qubits (Dual lattice)

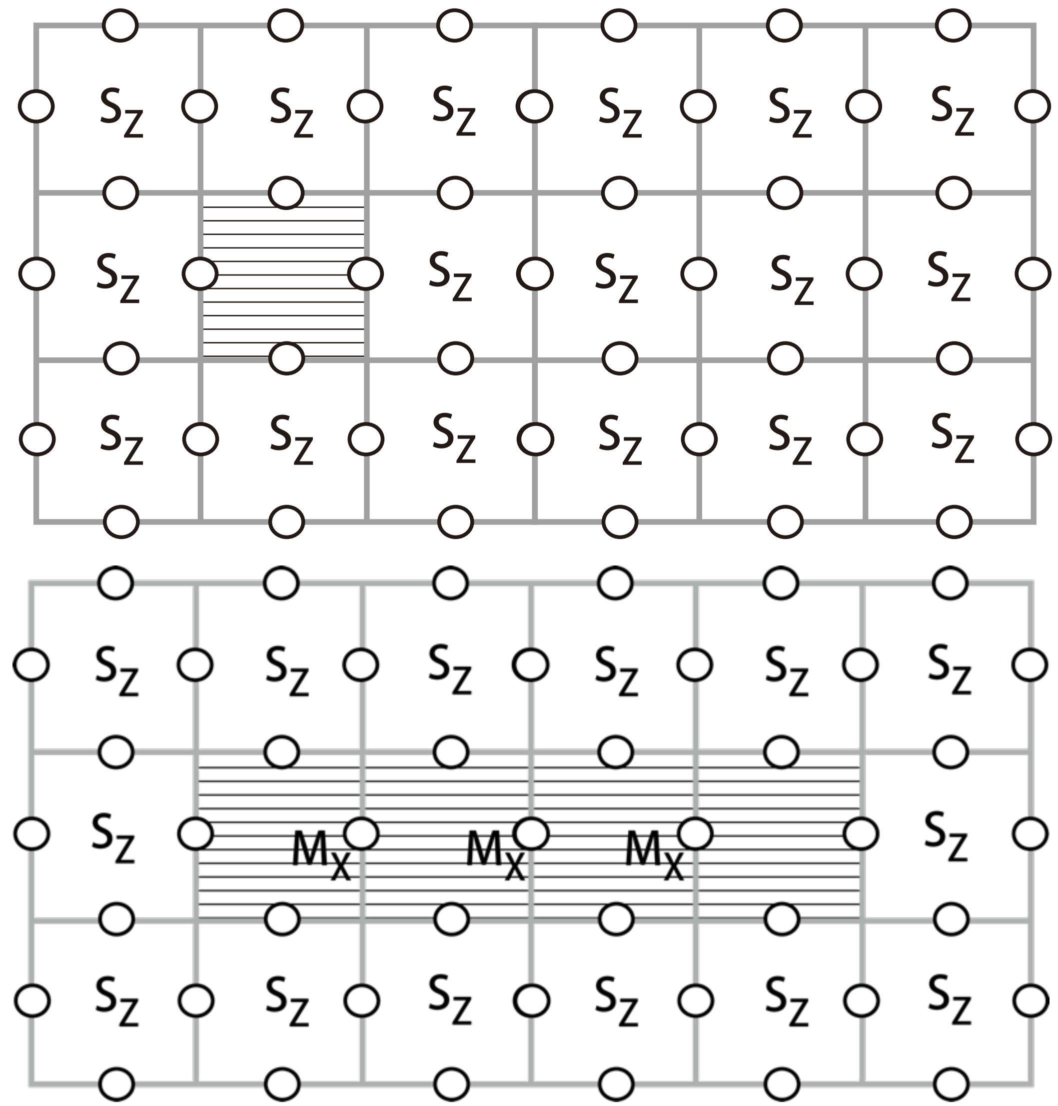


- Can make a different type of qubit
- X_L and Z_L reversed
- Primal and dual lattice

Moving holes



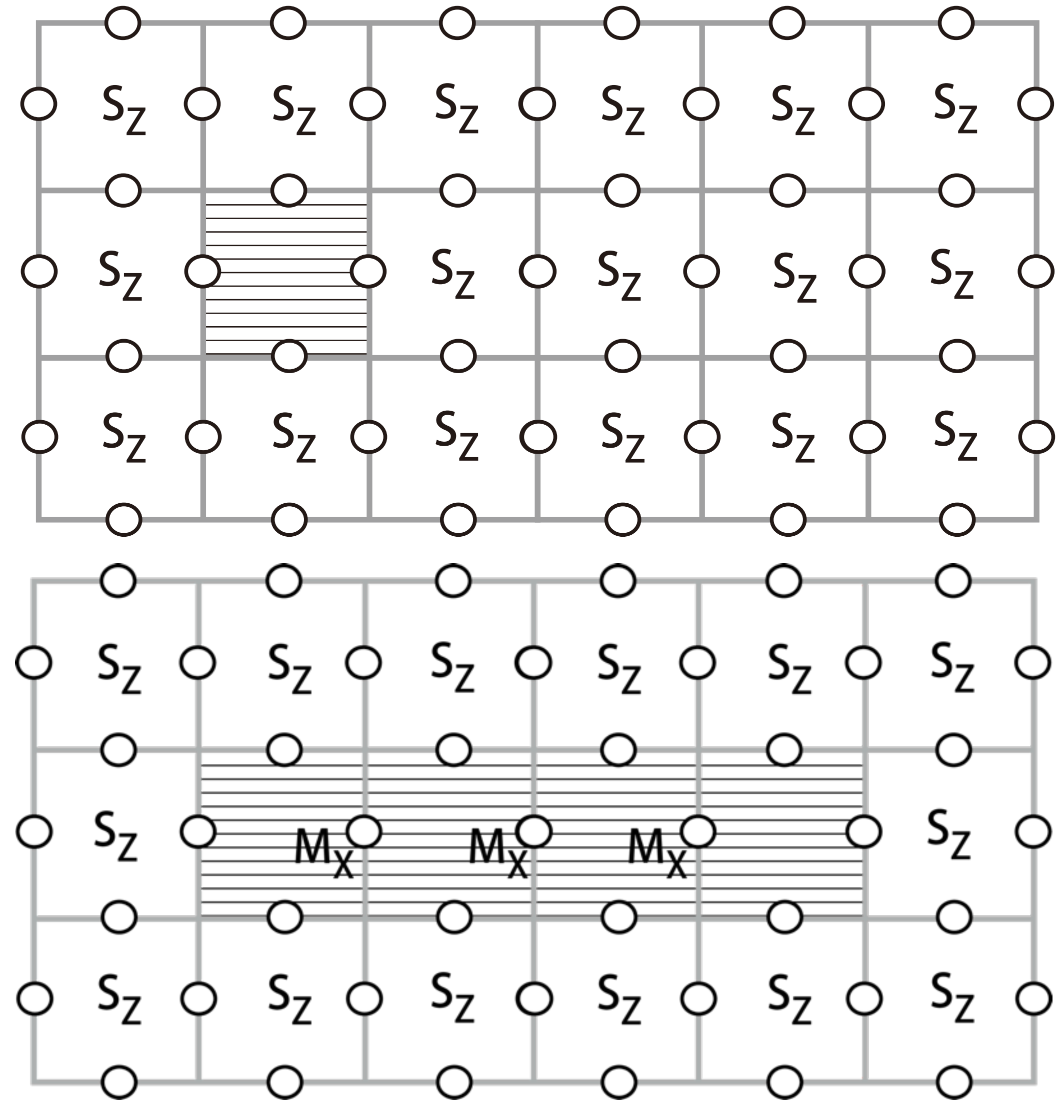
Moving holes



- First, expand hole using measurements and stop measuring stabilizers

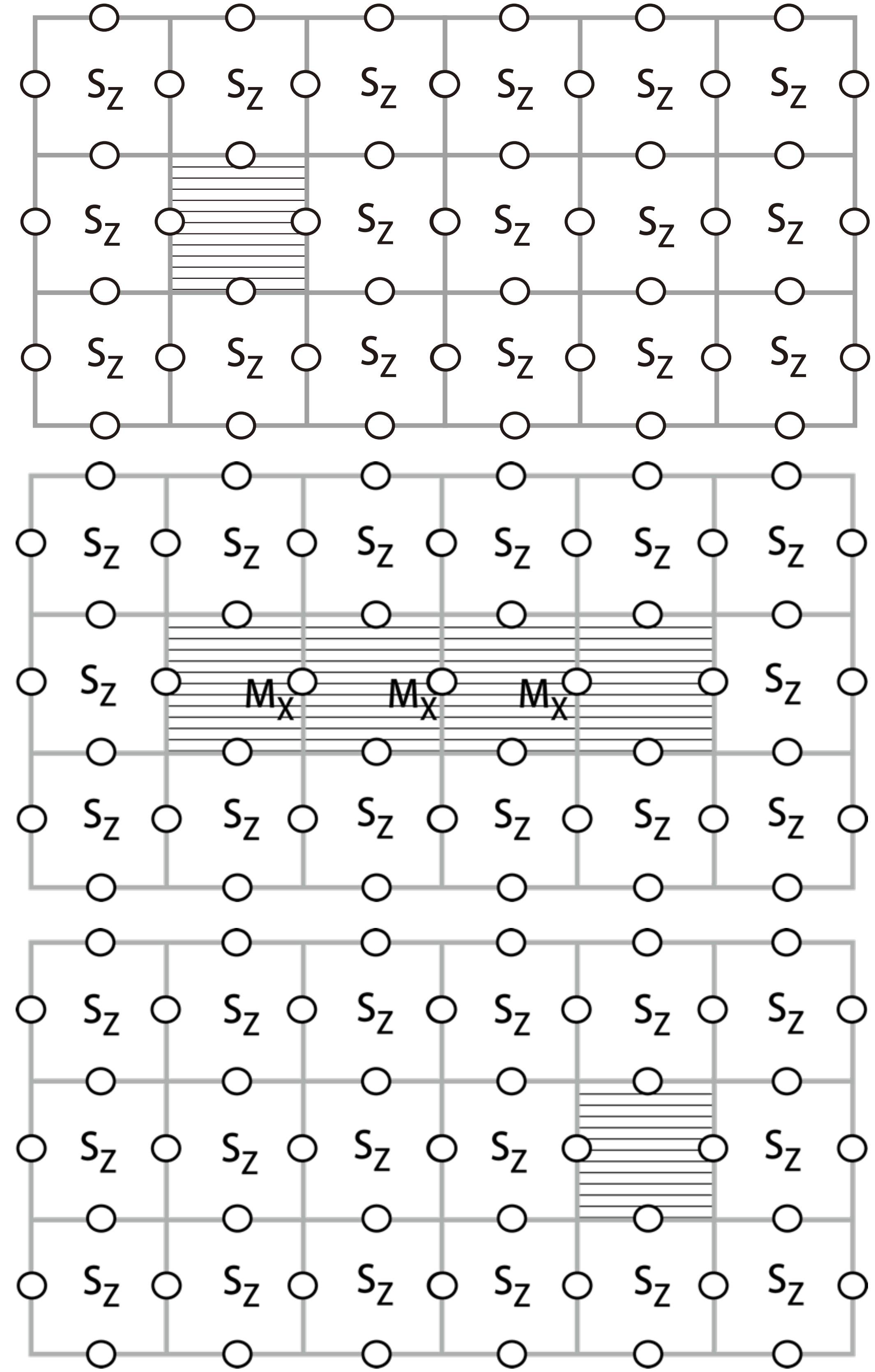
Moving holes

- First, expand hole using measurements and stop measuring stabilizers
- Fix up w/ bit flips (takes multiple rounds, depending on distance)

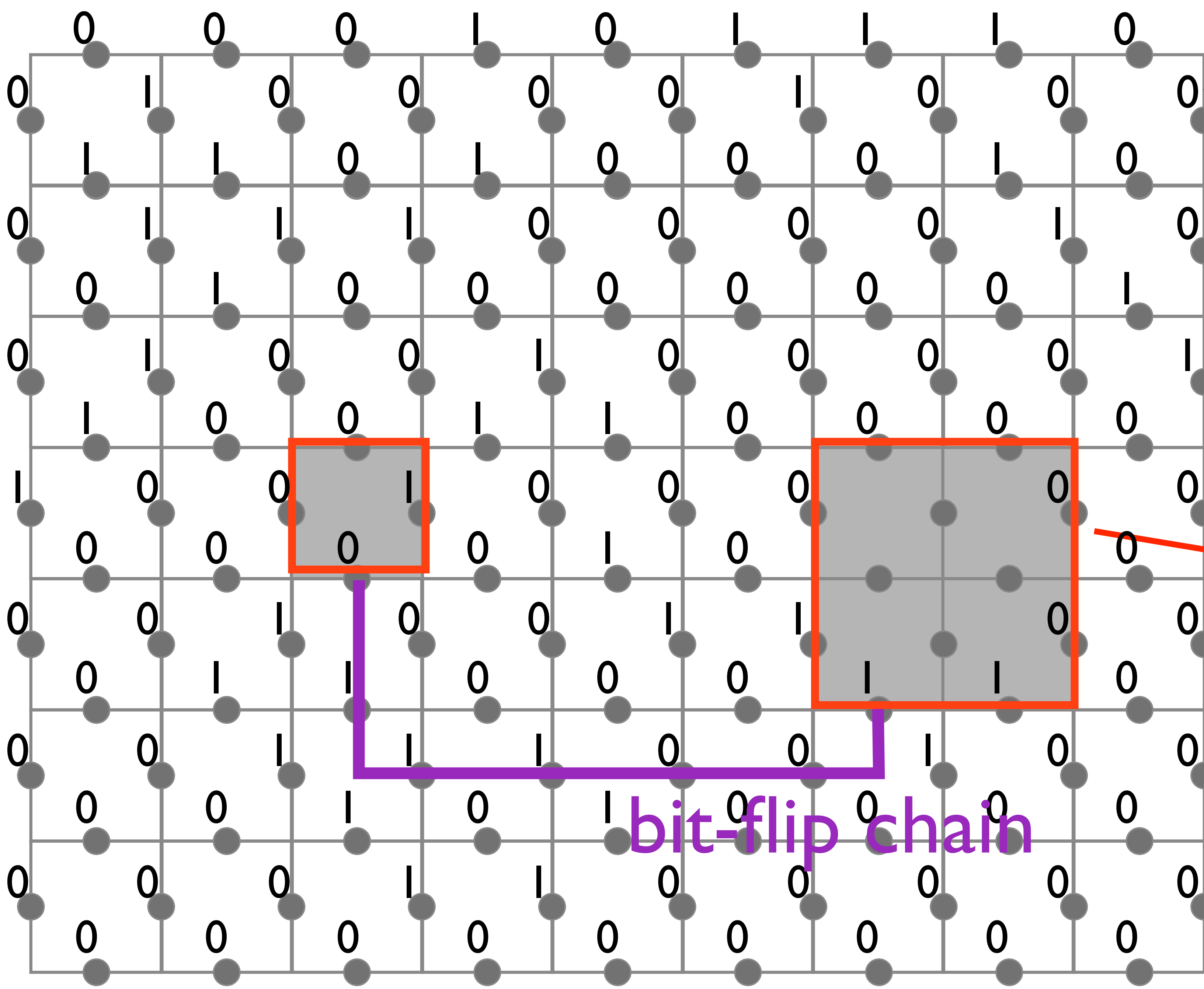


Moving holes

- First, expand hole using measurements and stop measuring stabilizers
- Fix up w/ bit flips (takes multiple rounds, depending on distance)
- Shrink hole to new location



Simplest Example
~NOT gate operation~



measured eigenvalue

data qubit

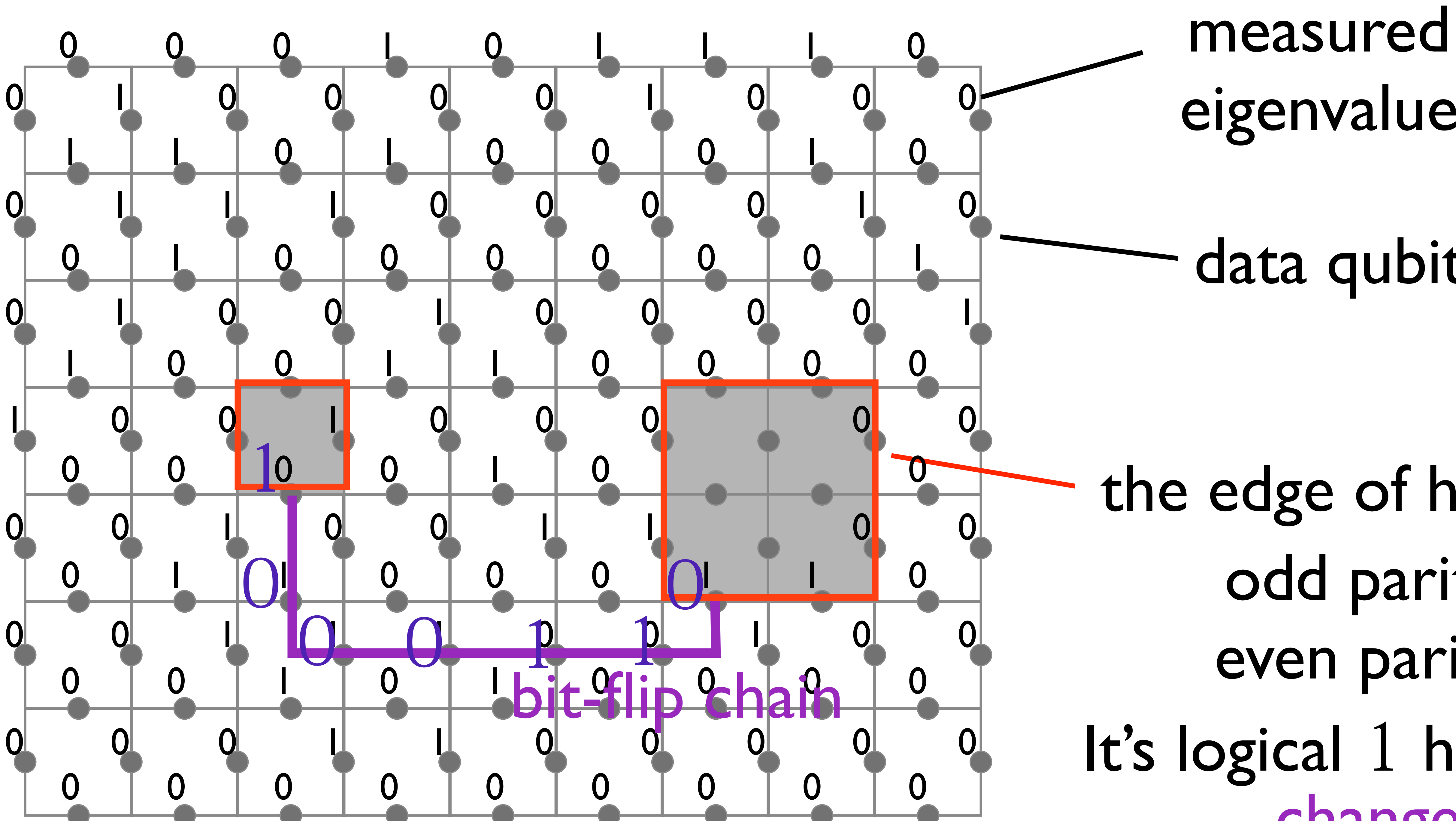
the edge of hole

odd parity: 0

even parity: 1

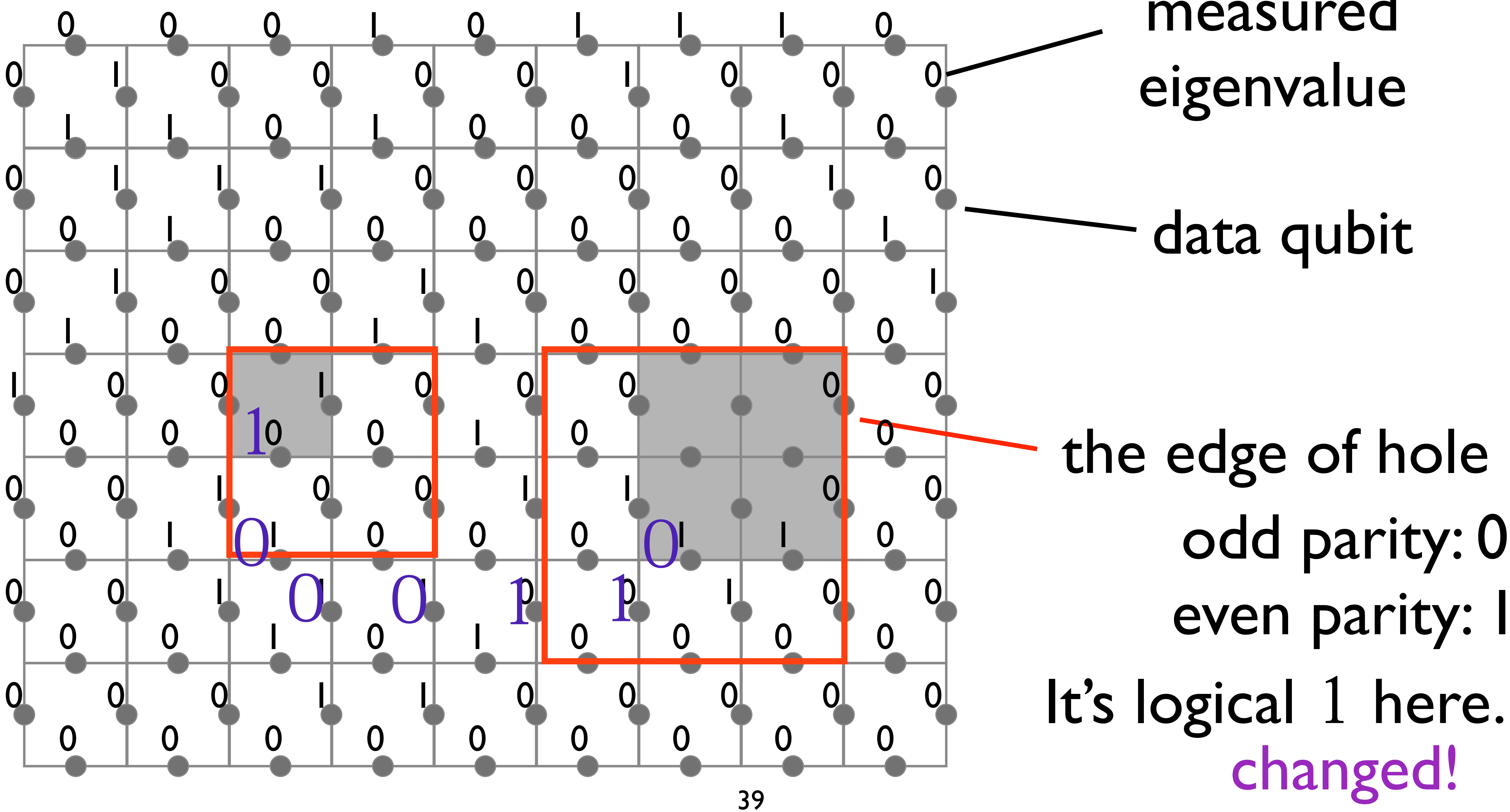
It's logical 0 here.

Simplest Example
~NOT gate operation~

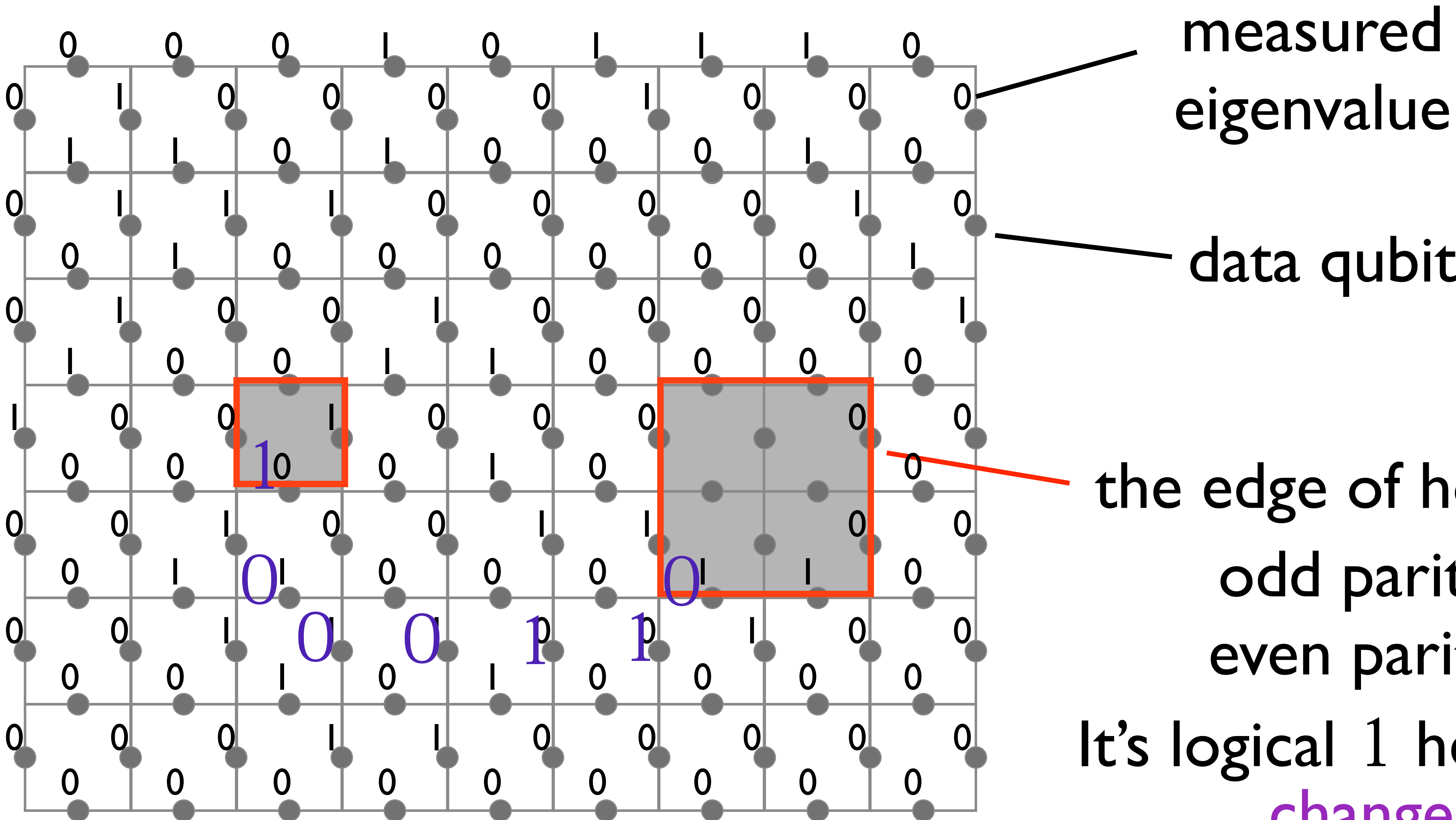


odd parity: 0
even parity: 1
It's logical 1 here.
changed!

Simplest Example
~NOT gate operation~



Simplest Example
~NOT gate operation~



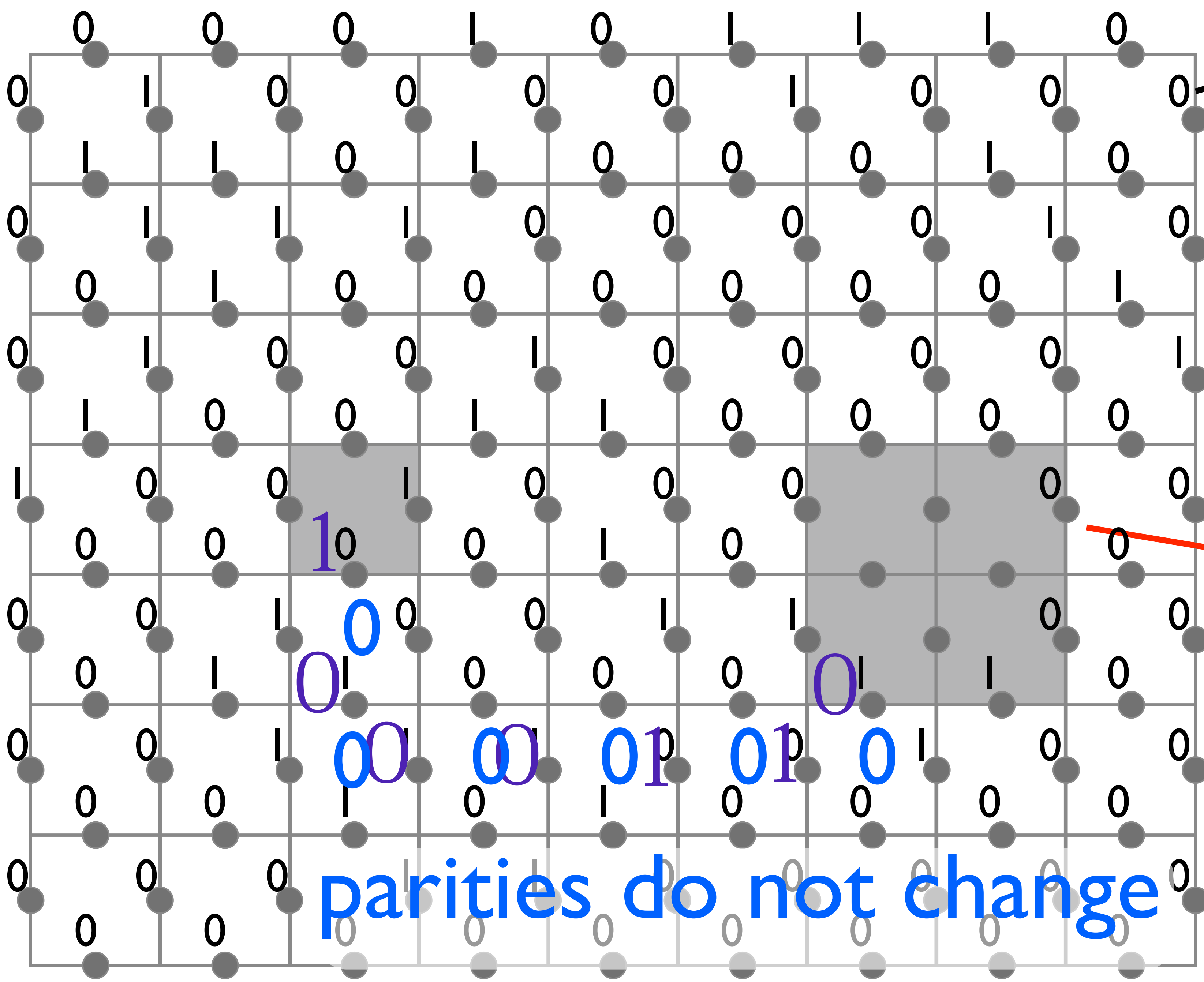
odd parity: 0

even parity: 1

It's logical 1 here.

changed!

Simplest Example
~NOT gate operation~



measured eigenvalue

data qubit

the edge of hole

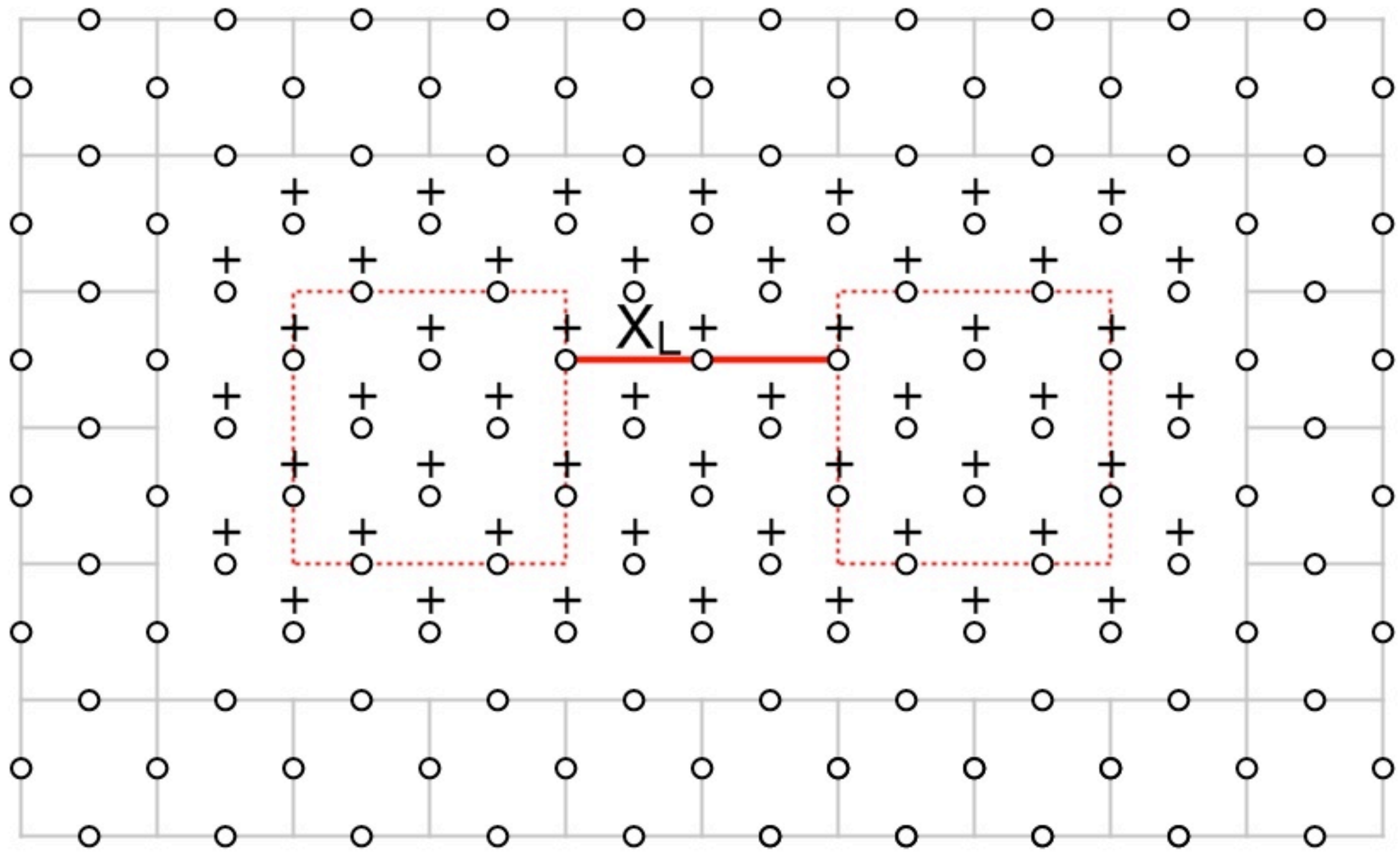
odd parity: 0

even parity: 1

It's logical 1 here.

changed!

Initializing logical qubits



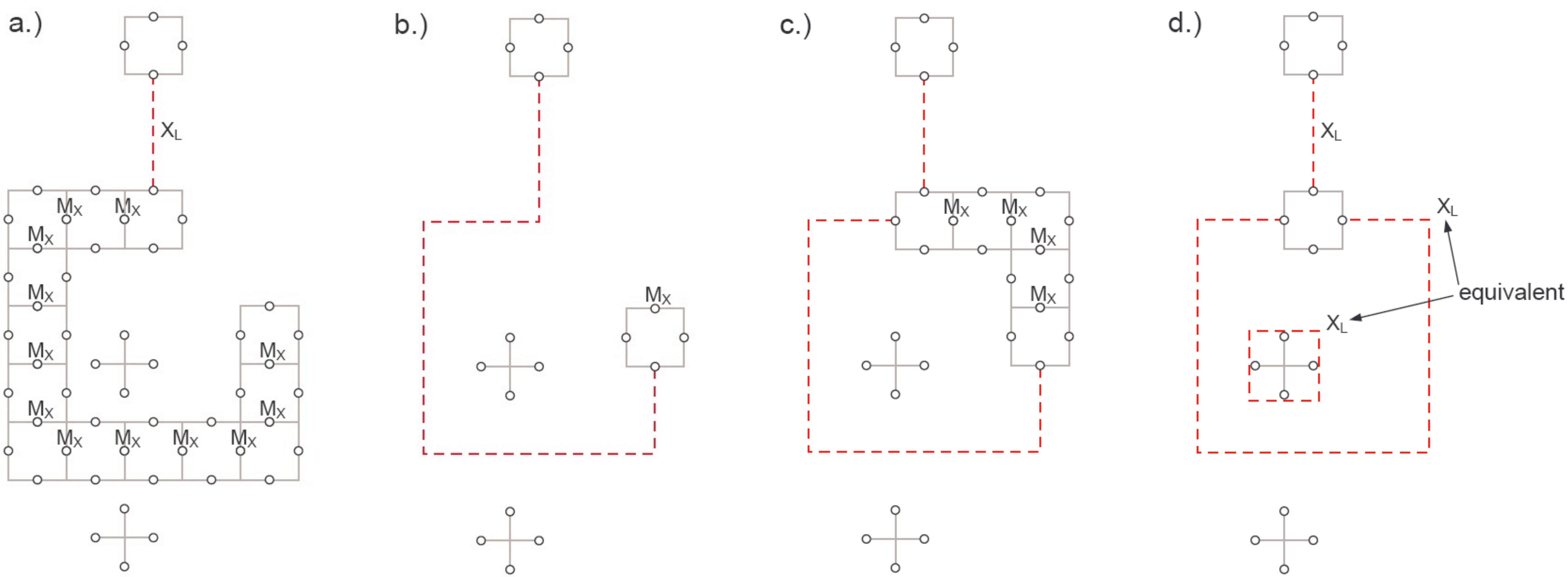
- Start with region of $|+\rangle$
- Smooth qubit in ±1 eigenstate of X_L
- Measure Z stabilizers

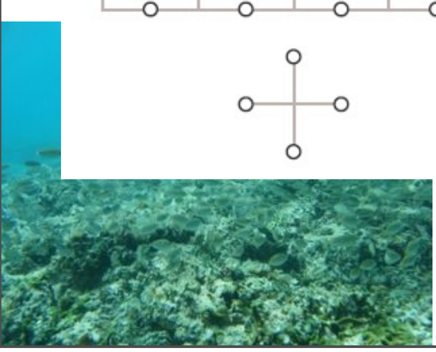
42

Braiding a CNOT

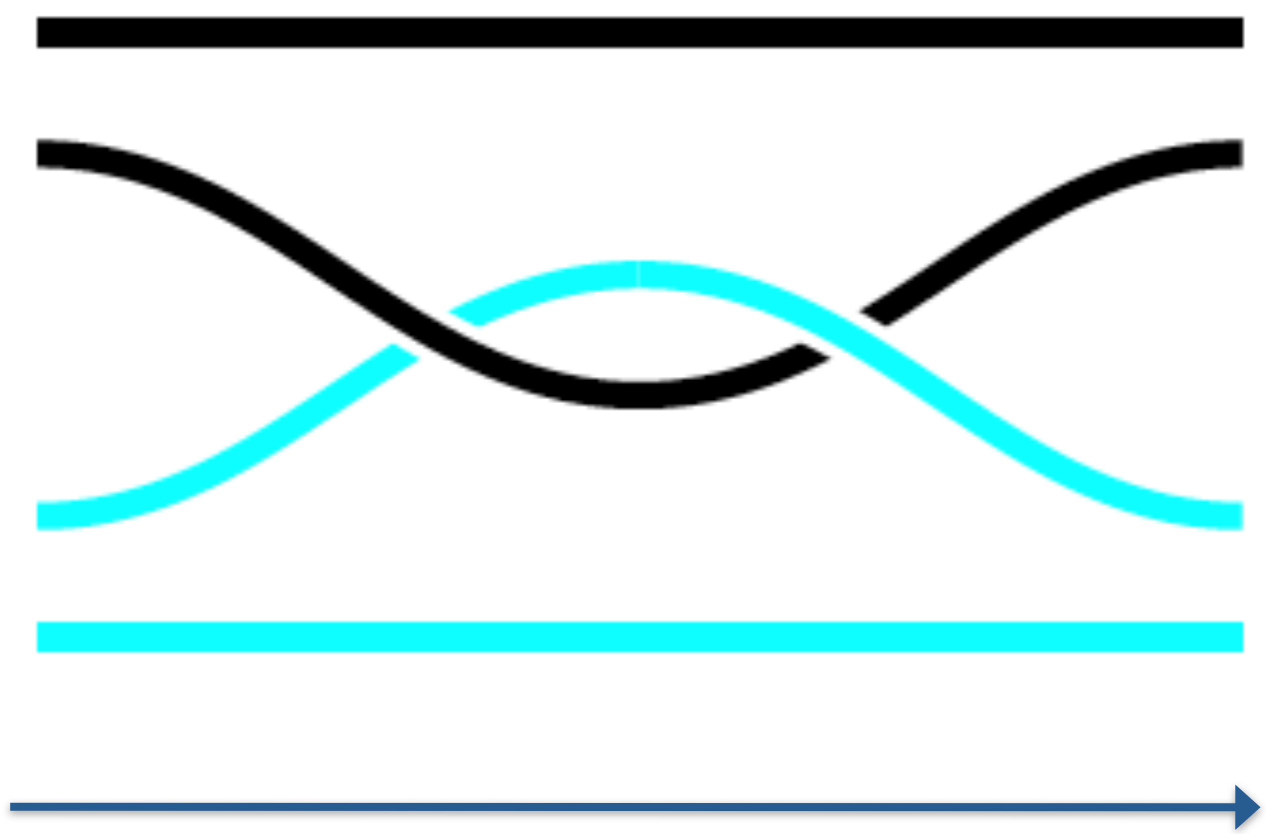
- Braiding smooth pair with rough pair gives CNOT

  (WHY? Stay tuned...)





Braiding: 2D+T Picture



} dual qubit.

(Rough holes)

} primal qubit.

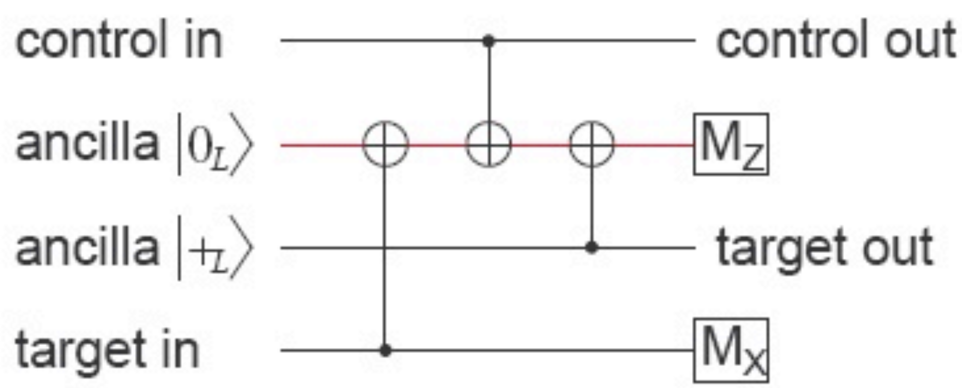
(Smooth holes)

Time

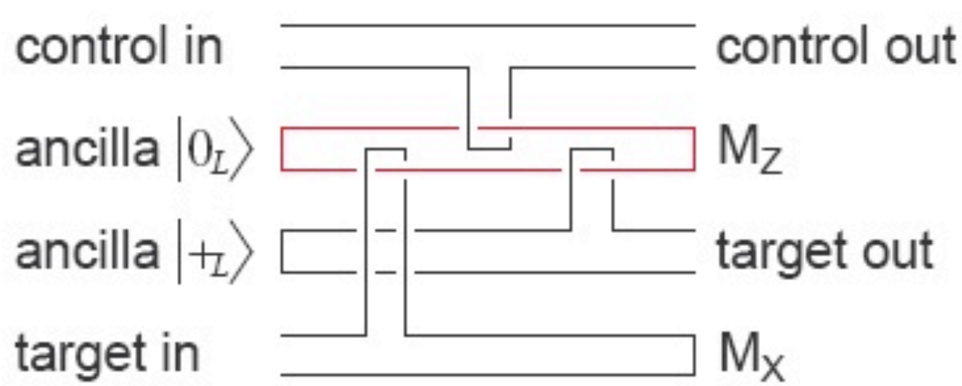
picture from Raussendorf et al., NJP 9, 2007

Smooth-smooth CNOT

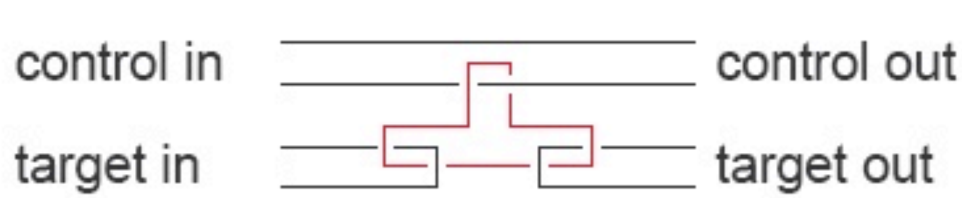
- Circuit is equivalent to CNOT followed by $(Z \otimes Z)^{M_X}$ followed by $(X_t)^{M_Z}$



- Circuit can be represented as a braiding of defects



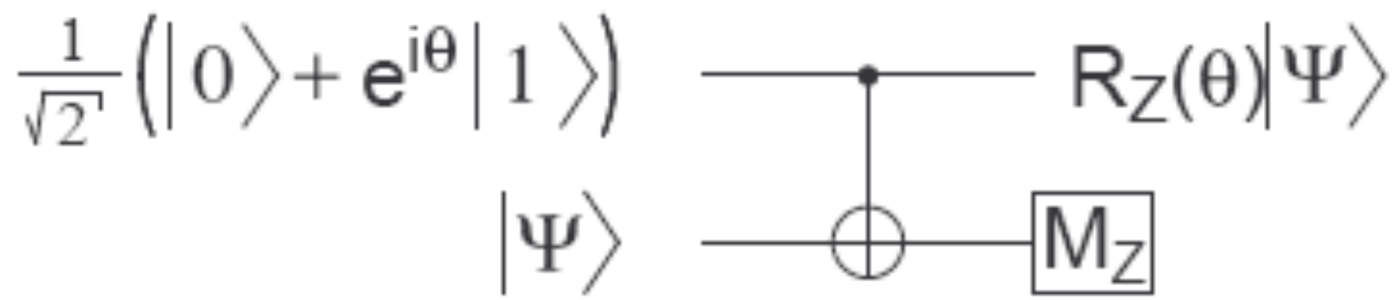
- Equivalent more compact braiding



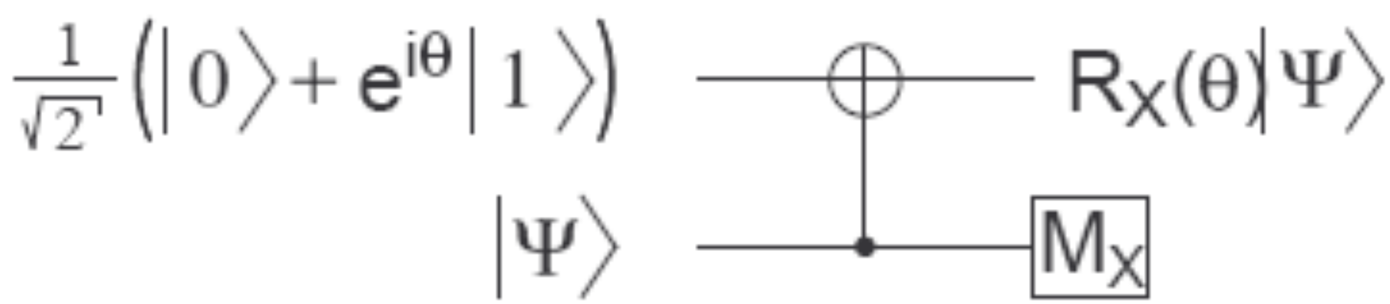
Non-Clifford gates

- We can prepare states $|0_L\rangle$, $|1_L\rangle$, $|+_L\rangle$ and $|-_L\rangle$
- We can perform gates X_L, Z_L, M_X, M_Z and CNOT
- Not universal without the following:

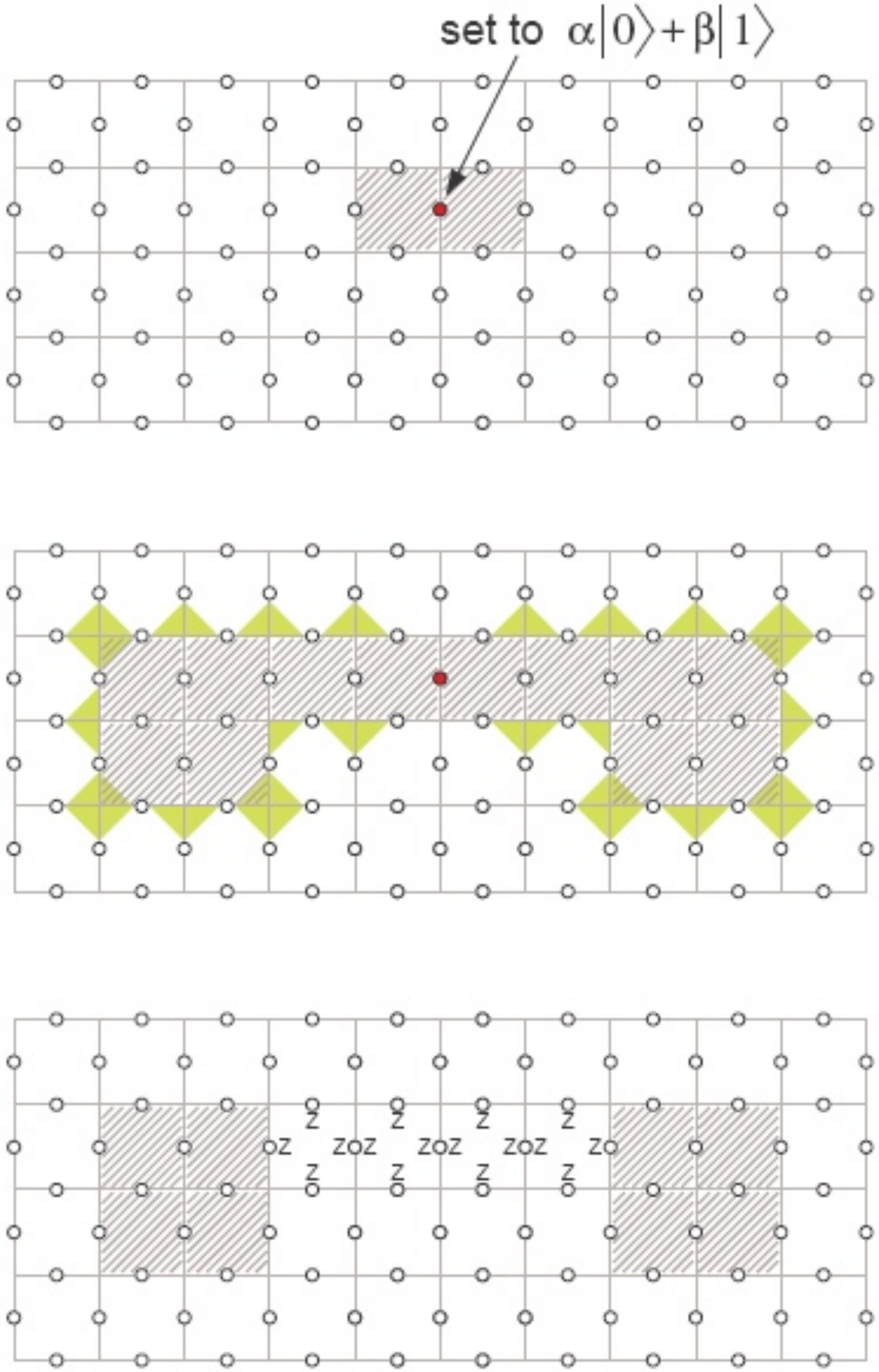
a.)



b.)



State injection

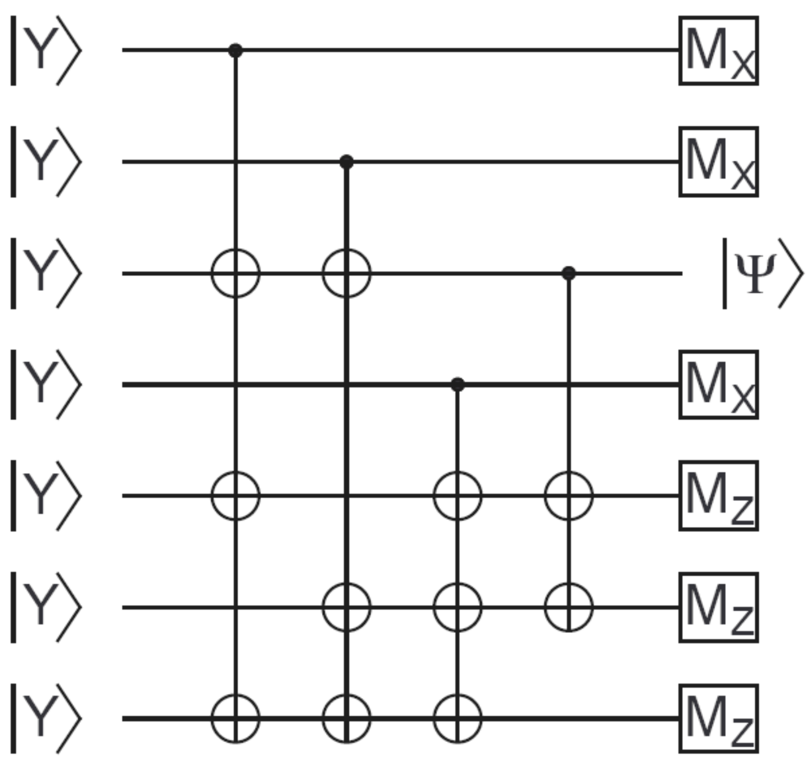


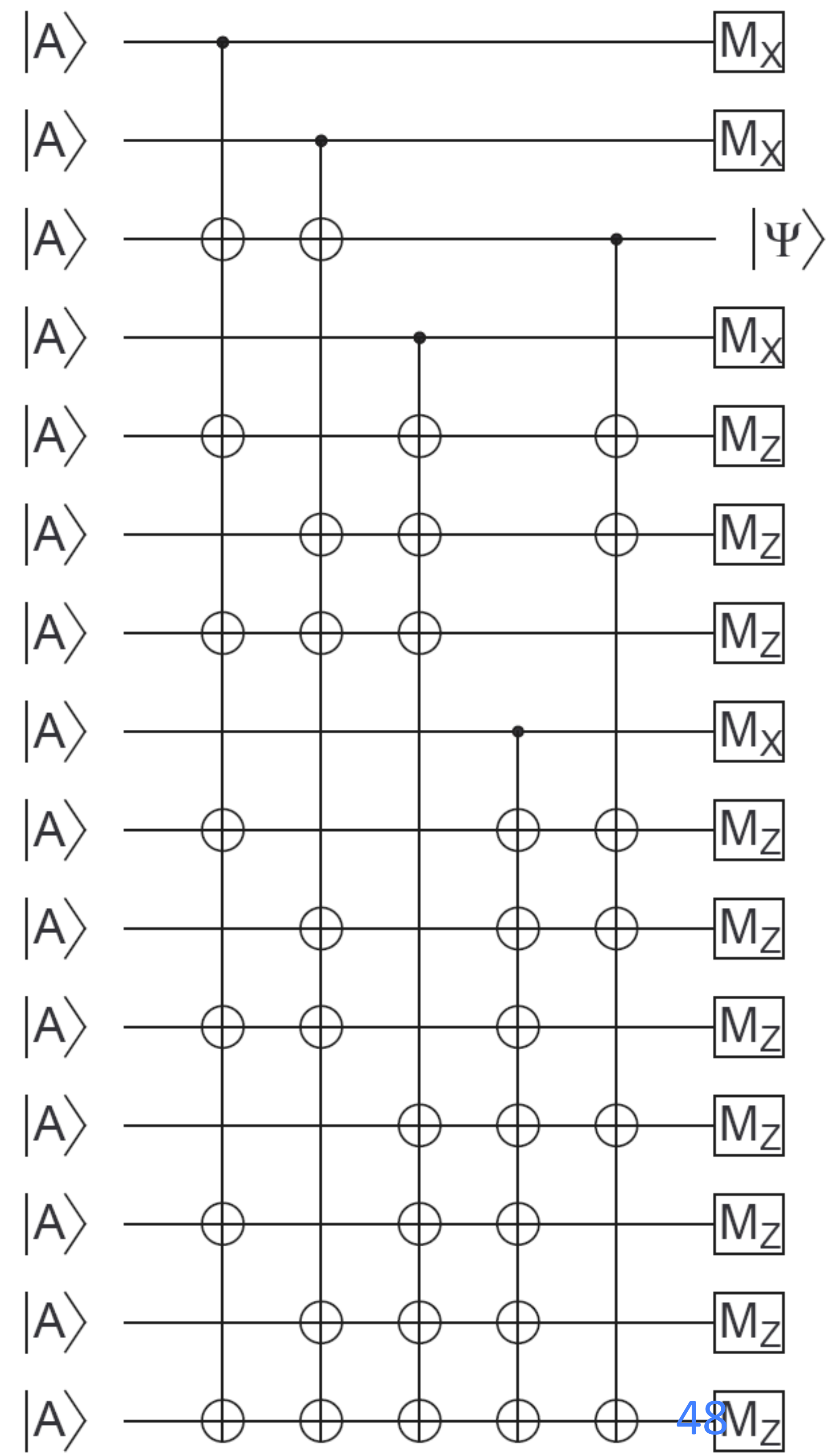
- Set a single qubit to desired state $\alpha|0\rangle + \beta|1\rangle$ (M_Z then rotate)
- Increase size of defects using X measurements
- Separate defects by measuring and correcting Z stabilizes
- Procedure is not fault-tolerant, restart if errors detected early
- Logical state will not be perfect $\alpha|0\rangle + \beta|1\rangle$, however...

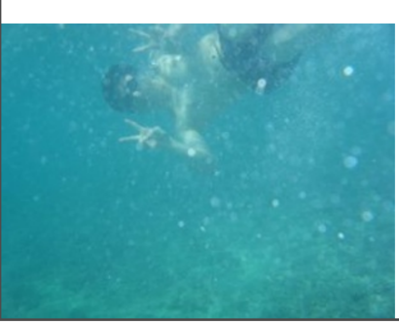
Aqua : Advancing Quantum Architecture

State distillation: “Singular factories”

- Two very special states exist $|Y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ and $|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$
- These states can be “distilled”
- Imperfect $|Y\rangle$ and $|A\rangle$ states approach perfect versions exponentially quickly
- Probability of success asymptotically close to 1, though some byproduct operators needed







48

Universality

- Adding these states:

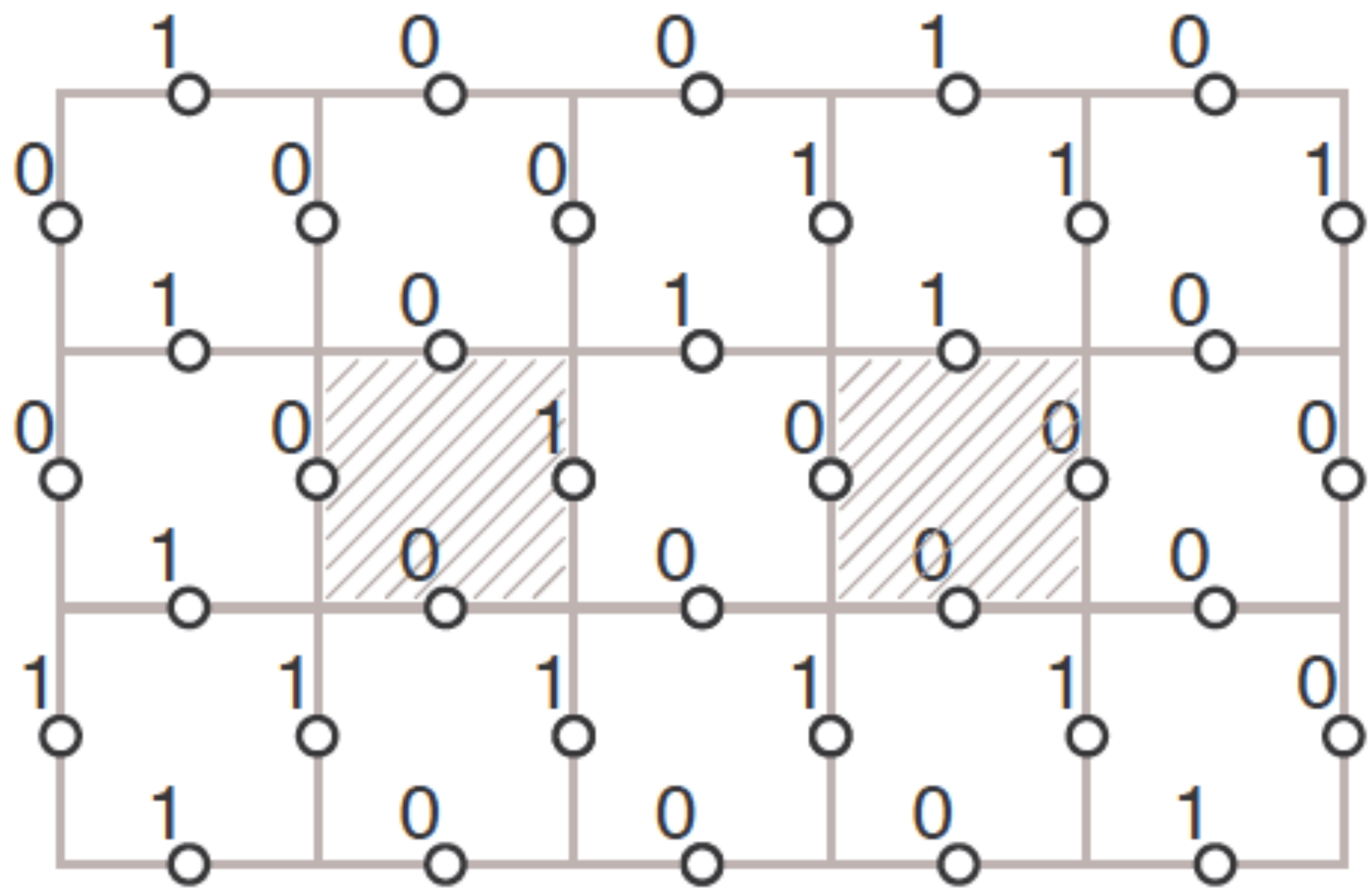
$$|Y\rangle := (|0\rangle + \mathrm{i}|1\rangle)/\sqrt{2}$$

$$|A\rangle := (|0\rangle + \mathrm{e}^{\mathrm{i}\pi/4}|1\rangle)/\sqrt{2}$$

gives us *probabilistic, heralded* universality

- Logical Toffoli gate consumes 7 $|Y\rangle$ and 4.5 $|A\rangle$ states (average)
- Distillation of enough states for *one* Toffoli gate ~1800 singular qubits + braidings (certain assumptions, not detailed here)

Measuring a Qubit



- Measure region in Z (or X) basis
- Every ring around either defect odd parity, state is 1
- Majority vote when not all same

CALTECH

CALIFORNIA INSTITUTE OF TECHNOLOGY 1891

USC Viterbi School of Engineering

USC UNIVERSITY OF SOUTHERN CALIFORNIA

KEIO UNIVERSITY 1858 CALAMVS GLADIO FORTIOR

52

自己紹介

ISI
Information Sciences Institute
USC Viterbi School of Engineering

NOKIA
Connecting People

ASACA

ShibaSoku
high quality broadcast and video equipment

MOSIS

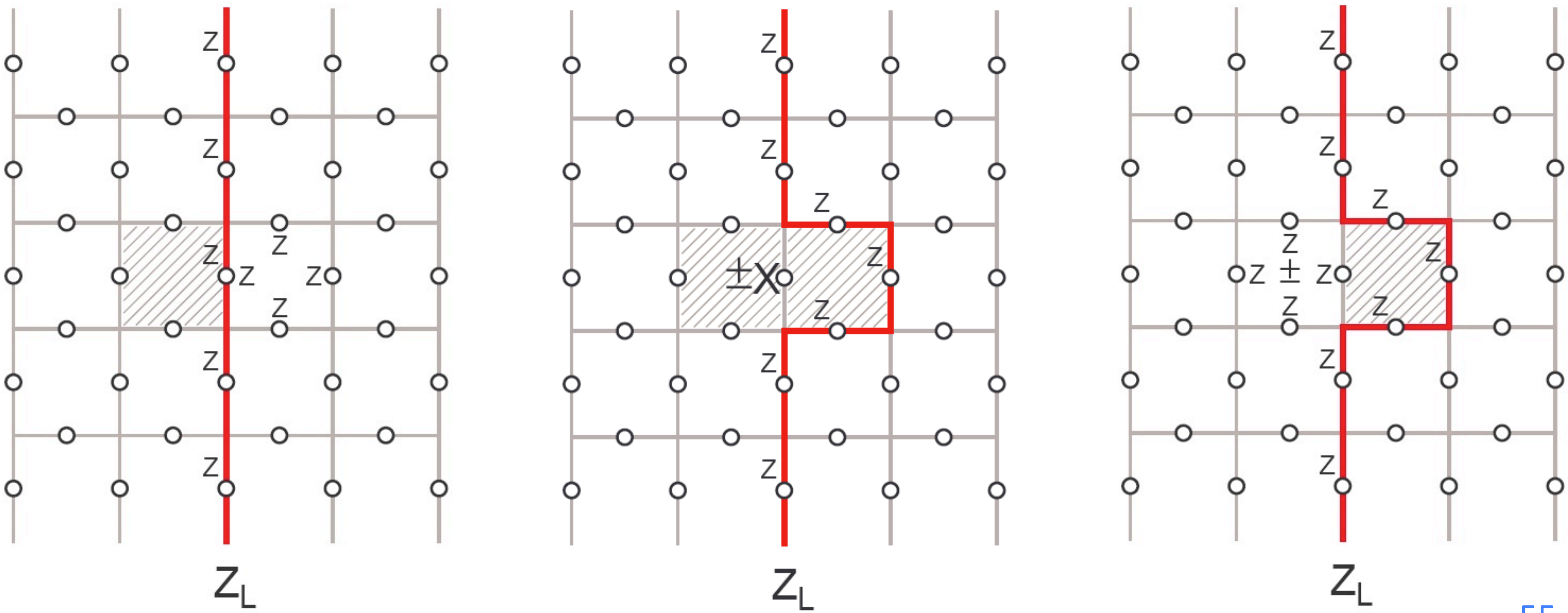
Quantum.

Theory

- Why braiding works
- Code distance
- Estimating logical error rate

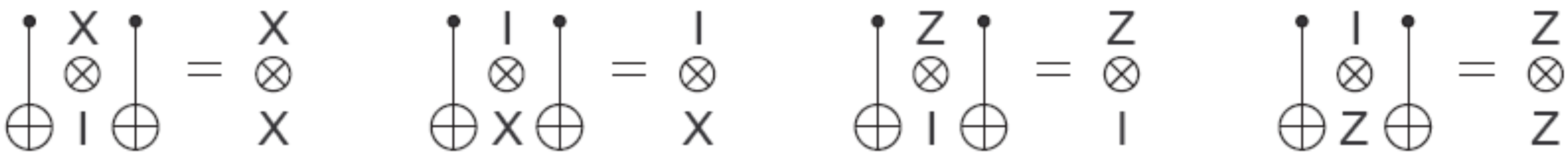
CNOT: moving defects

- Start with surface $|\Psi\rangle$ satisfying $Z_L|\Psi\rangle = |\Psi\rangle$ and Z_{face}
- Measure center qubit in the X basis to produce $|\Psi'\rangle$
- Surface $|\Psi'\rangle$ satisfies $Z_L Z_{\text{face}}|\Psi'\rangle = |\Psi'\rangle$ and $\pm X|\Psi'\rangle = |\Psi'\rangle$

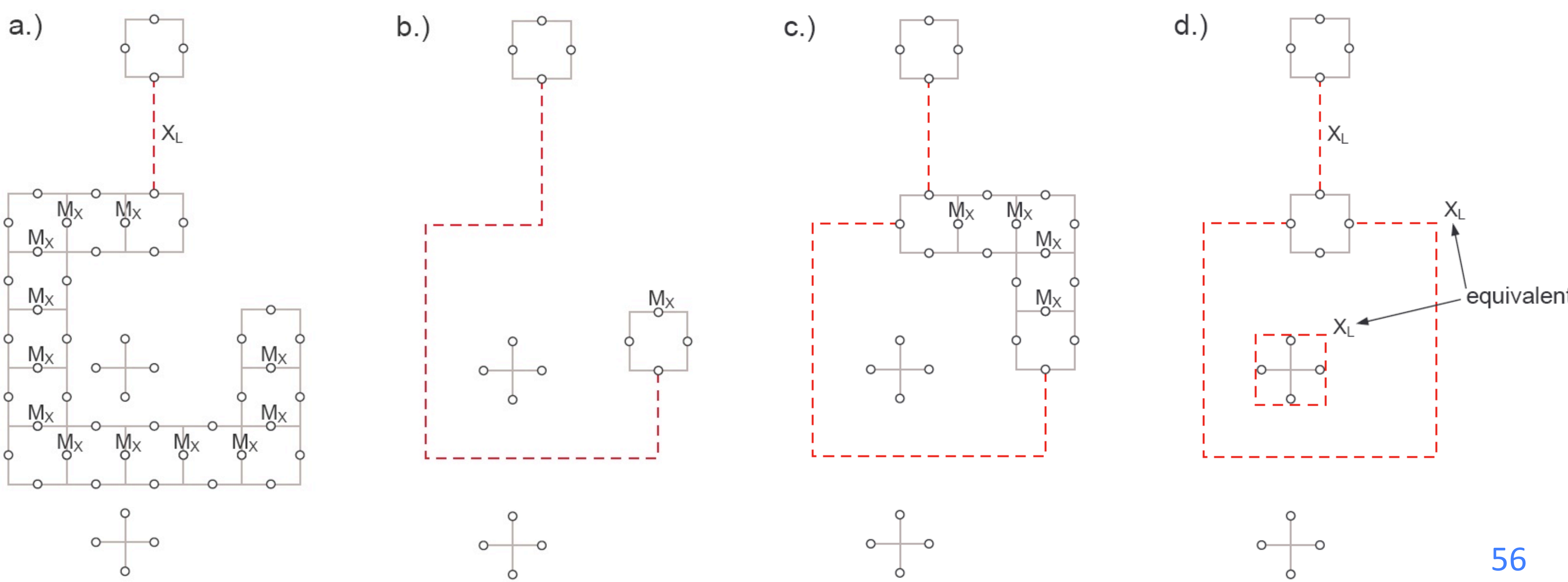


CNOT: $X \otimes I \rightarrow X \otimes X$

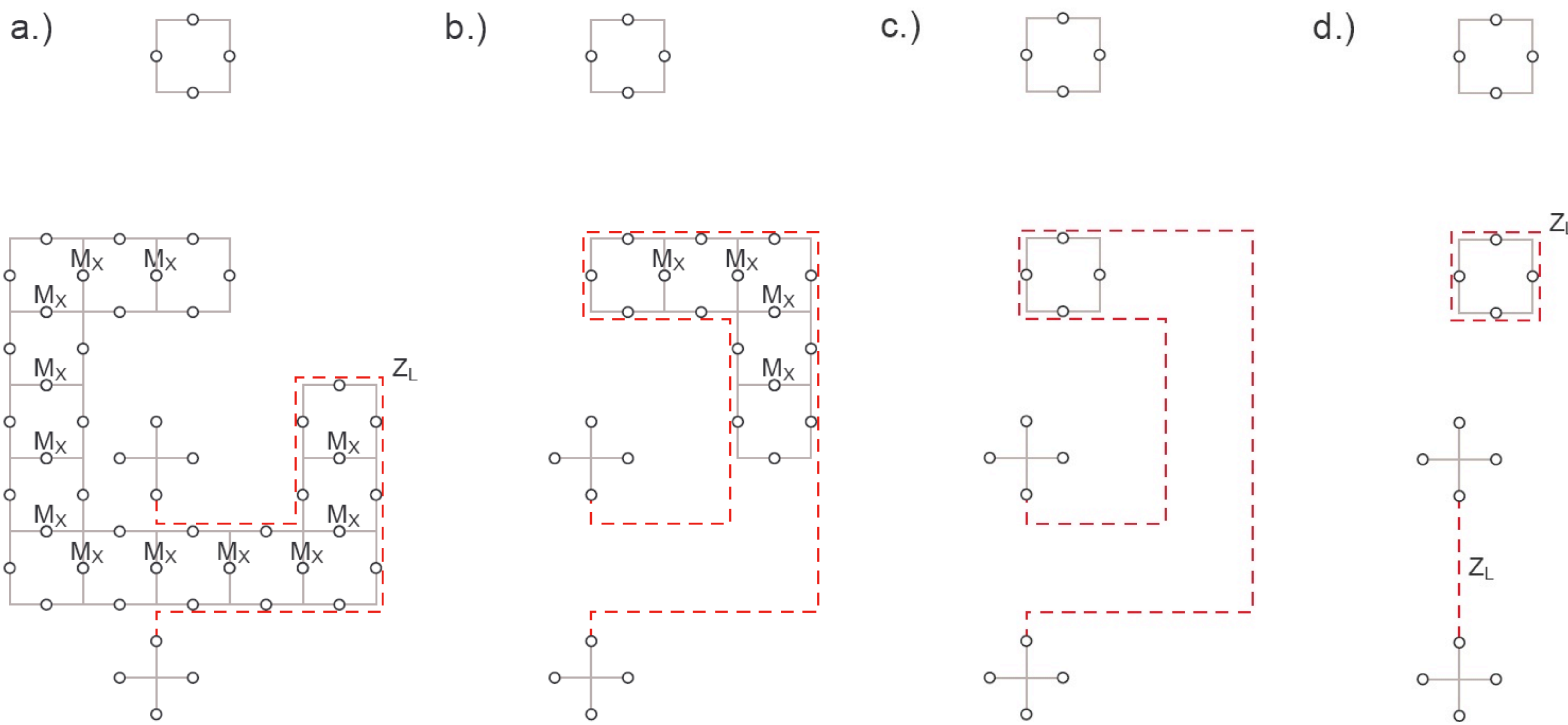
- If $M|\psi\rangle = |\psi\rangle$, then $U|\psi\rangle = UMU^{\dagger}U|\psi\rangle \Rightarrow M \rightarrow UMU^{\dagger}$
- CNOT manipulates stabilizers in the following manner:



- Need to show we can do this by braiding defects



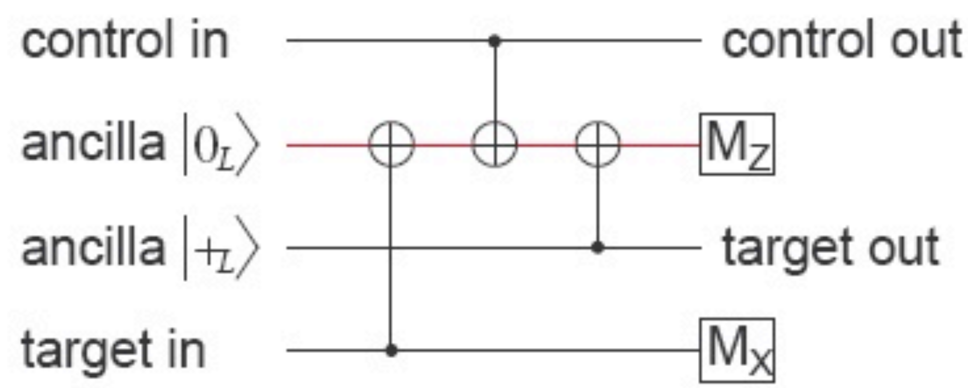
CNOT: $I \otimes Z \rightarrow Z \otimes Z$



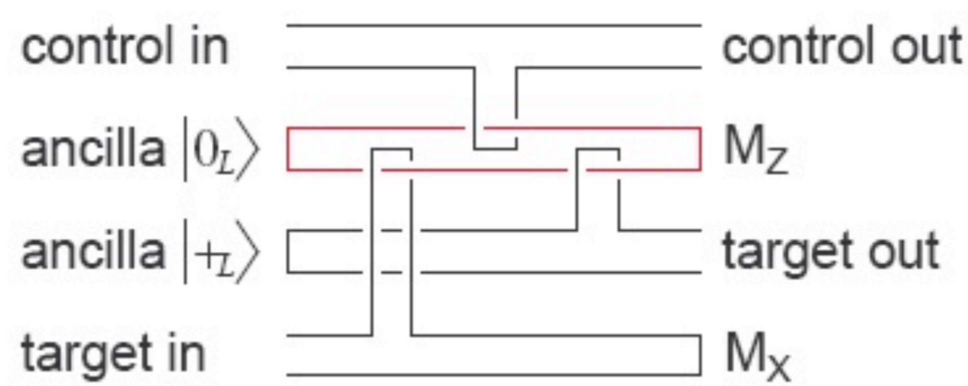
- Mappings $I \otimes X \rightarrow I \otimes X$ and $Z \otimes I \rightarrow Z \otimes I$ also easy to show
- Defects can be interacted over arbitrary distances in almost constant time
- Still need CNOT between two smooth qubits

Smooth-smooth CNOT

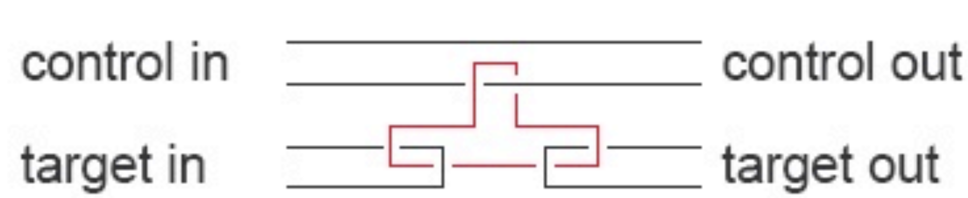
- Circuit is equivalent to CNOT followed by $(Z \otimes Z)^{M_X}$ followed by $(X_t)^{M_Z}$



- Circuit can be represented as a braiding of defects



- Equivalent more compact braiding

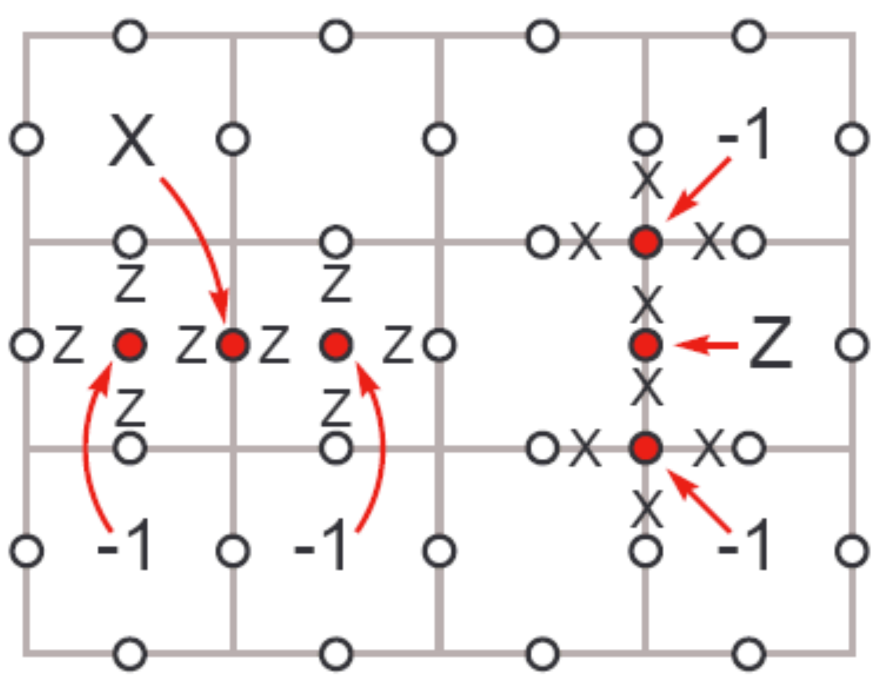


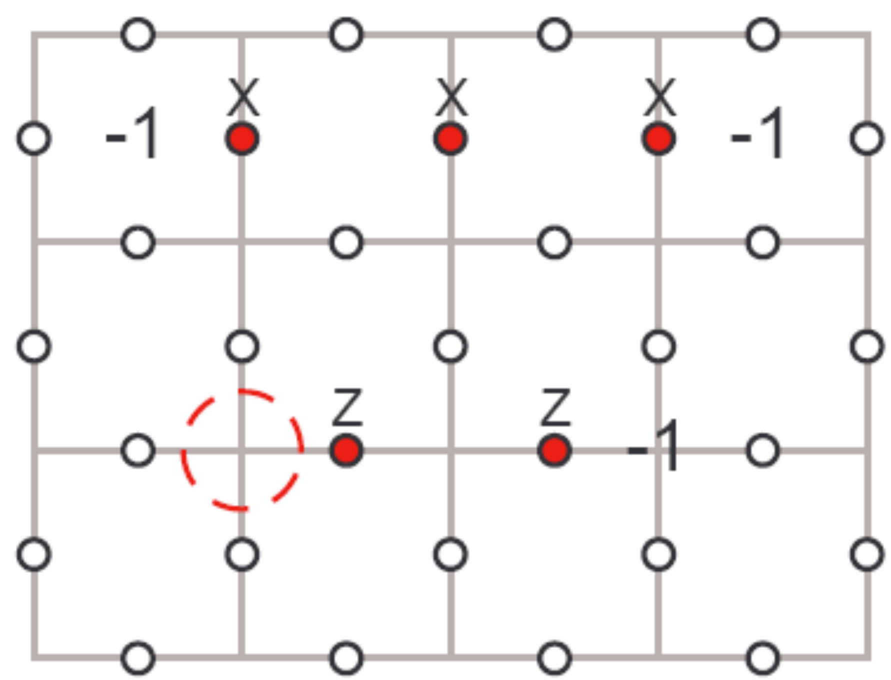
58

Error correction

- All error correction based on $XZ = -ZX$, eg:

$$Z_1 Z_2 Z_3 Z_4 |\psi\rangle = |\psi\rangle$$
$$Z_1 Z_2 Z_3 Z_4 X_1 |\psi\rangle = -X_1 Z_1 Z_2 Z_3 Z_4 |\psi\rangle = -X_1 |\psi\rangle$$

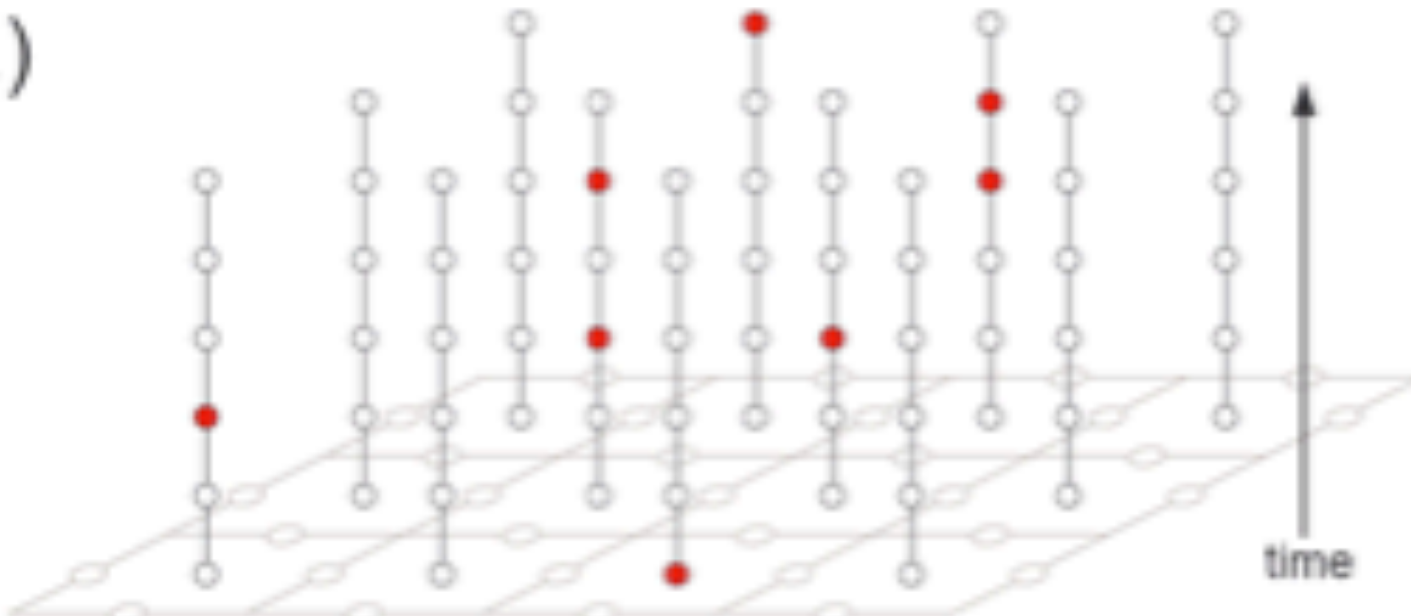




Need to carefully handle false syndromes and error chains

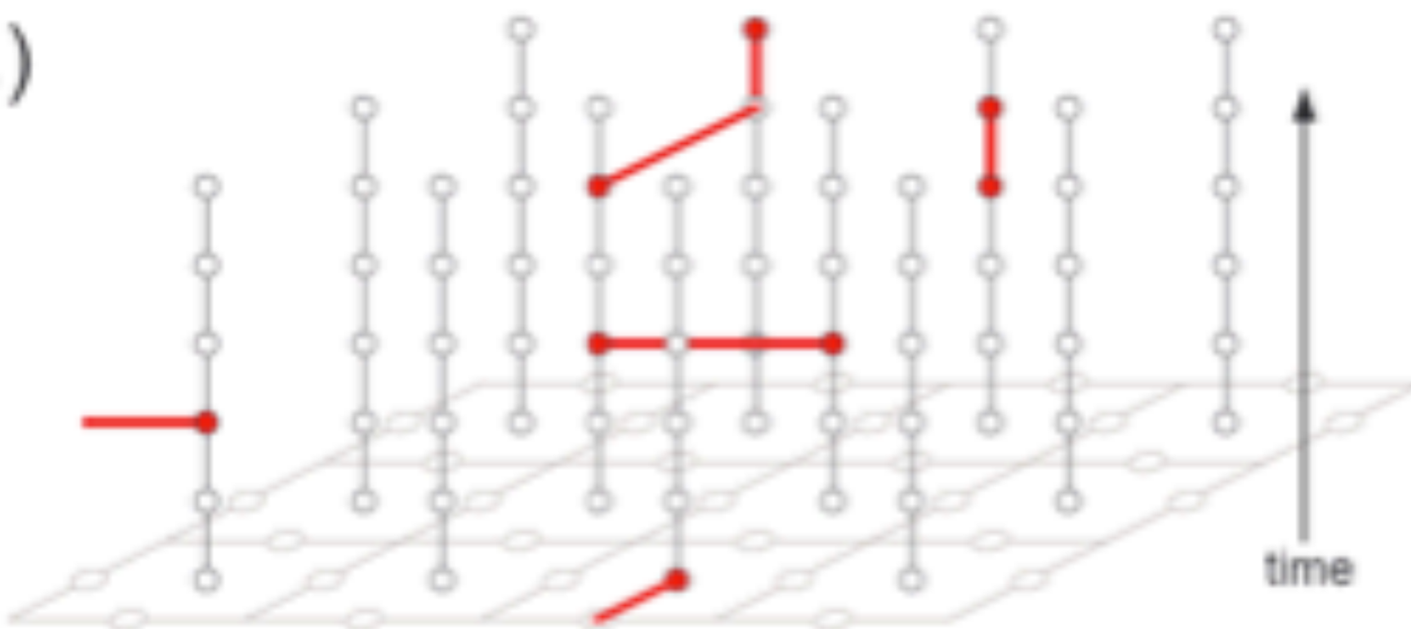
Error correction

a.)

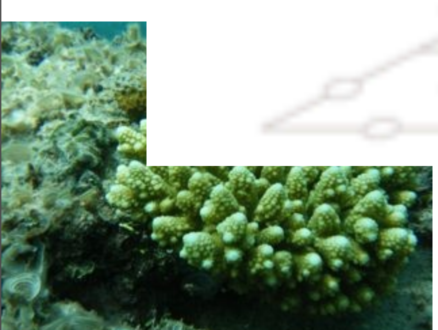


- Record time and position of changed syndromes

b.)

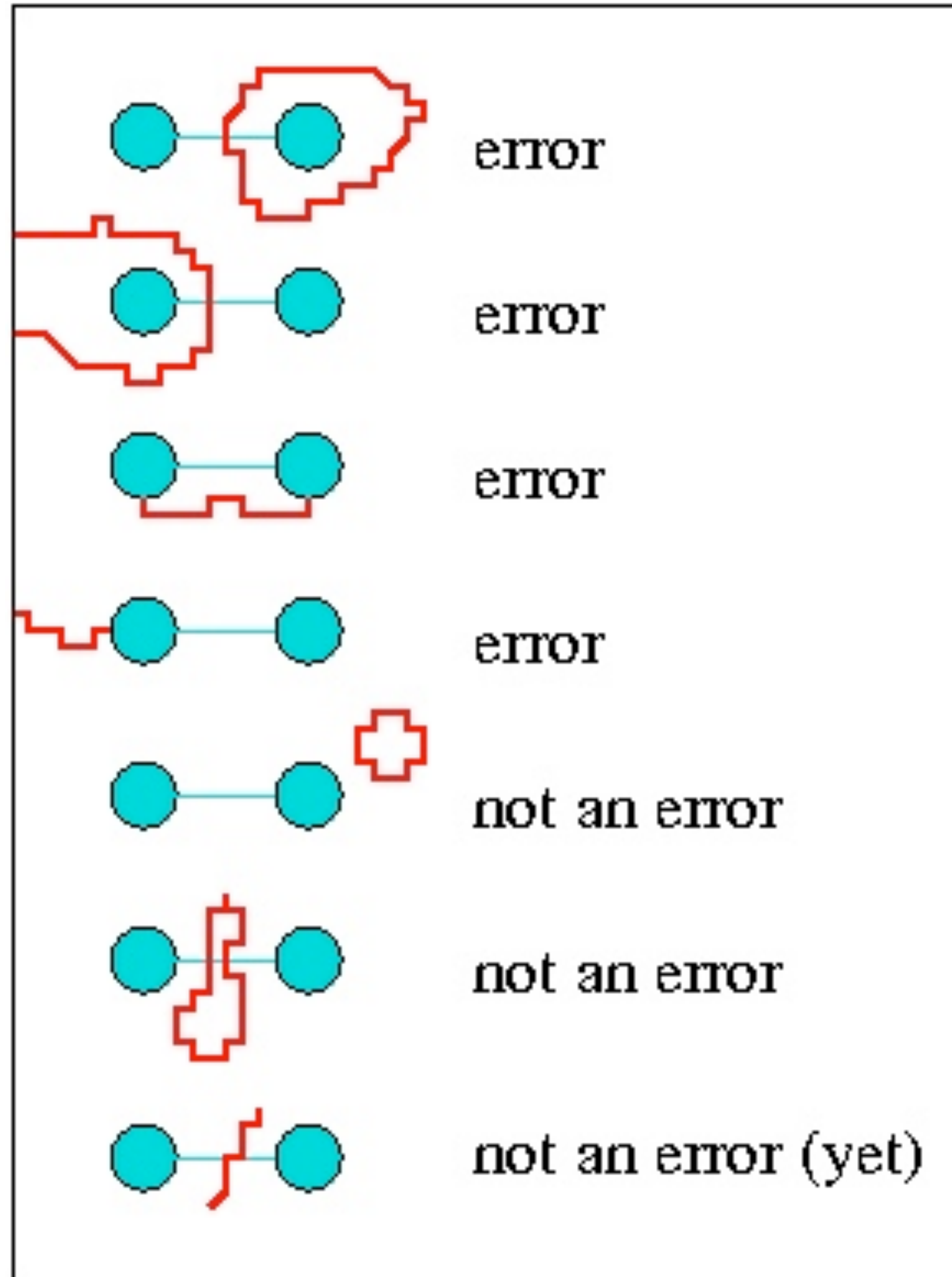


- Match closest pairs
- Apply corrective operations to spacelike edges
- Works *very* well, threshold error rate ~1%



Error Chains and Logical Gates

- Learn only endpoints of error chains
- Guess a path
- Usually (hopefully) results in meaningless loop
- Complete error chain (natural or mis-correct) results in logical X or Z gate



What is error correction?

62

What is error correction?

- To protect against errors, encode a few logical states in larger physical Hilbert space

What is error correction?

- To protect against errors, encode a few logical states in larger physical Hilbert space
- Physical errors take the system *out of the code space*
  - ➔ Goal of correction is to get back into code space

What is error correction?

- To protect against errors, encode a few logical states in larger physical Hilbert space
- Physical errors take the system *out of the code space*
  - ➔ Goal of correction is to get back into code space
- Some measure of "distance" from a legitimate code word (logical state)
  - ➔ Usually move to "closest" logical state

What is error correction?

- To protect against errors, encode a few logical states in larger physical Hilbert space
- Physical errors take the system *out of the code space*
  - ➔ Goal of correction is to get back into code space
- Some measure of "distance" from a legitimate code word (logical state)
  - ➔ Usually move to "closest" logical state
- More than halfway, you mis-correct
  - ➔ Mis-correct equivalent to accidentally executing a logical gate

Threshold

Threshold

- Executing error correction is itself a physical procedure

Threshold

- Executing error correction is itself a physical procedure
- EC can introduce errors

Threshold

- Executing error correction is itself a physical procedure
- EC can introduce errors
- *Threshold* is error level at which probability of logical error *declines* if you apply EC

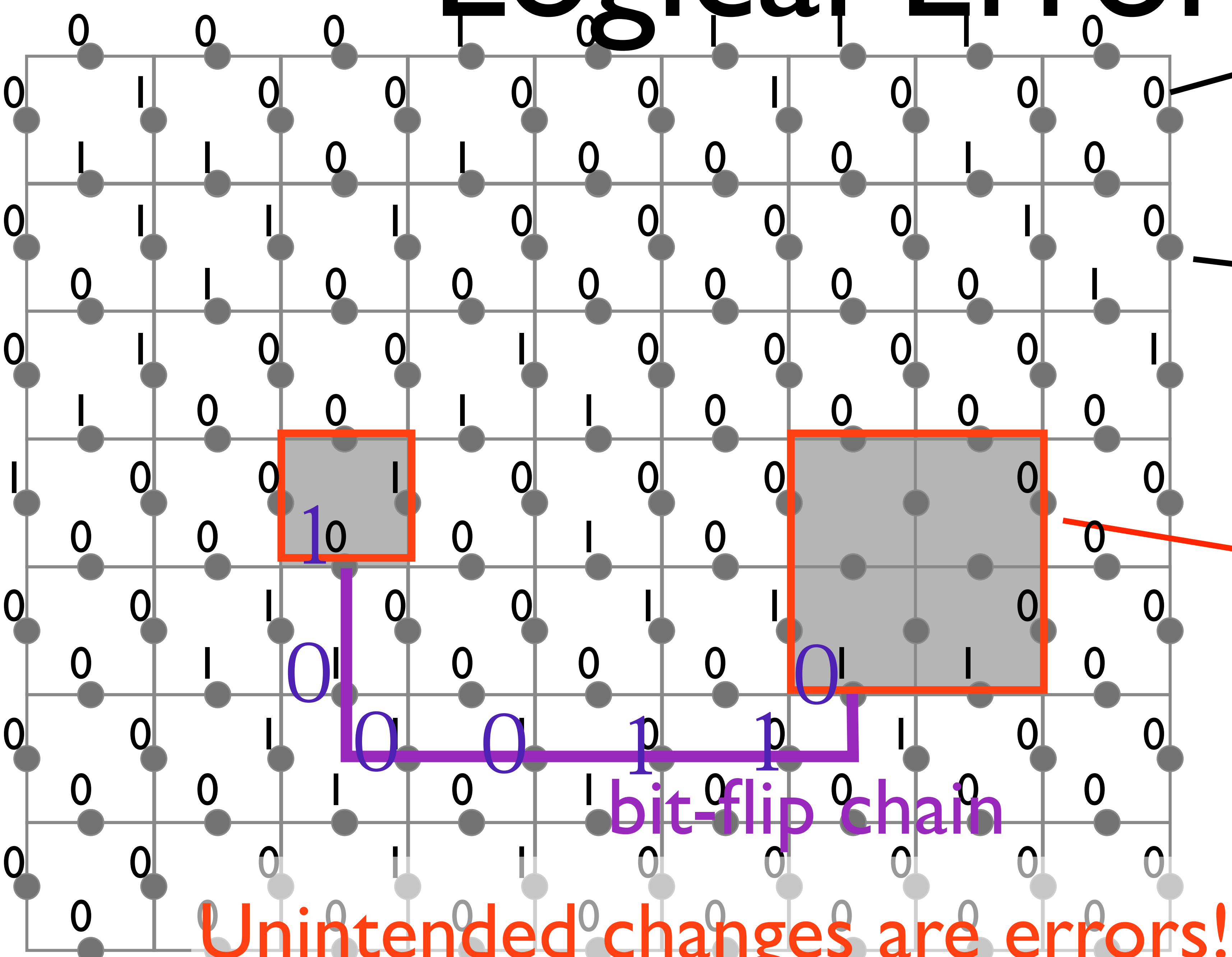
Threshold

- Executing error correction is itself a physical procedure
- EC can introduce errors
- *Threshold* is error level at which probability of logical error *declines* if you apply EC
- Most threshold analyses assume gate, memory, measurement errors same probability

Threshold

- Executing error correction is itself a physical procedure
- EC can introduce errors
- *Threshold* is error level at which probability of logical error *declines* if you apply EC
- Most threshold analyses assume gate, memory, measurement errors same probability
- Generally must *beat* threshold by 1-2 orders of magnitude

Simplest Example ~Logical Error~



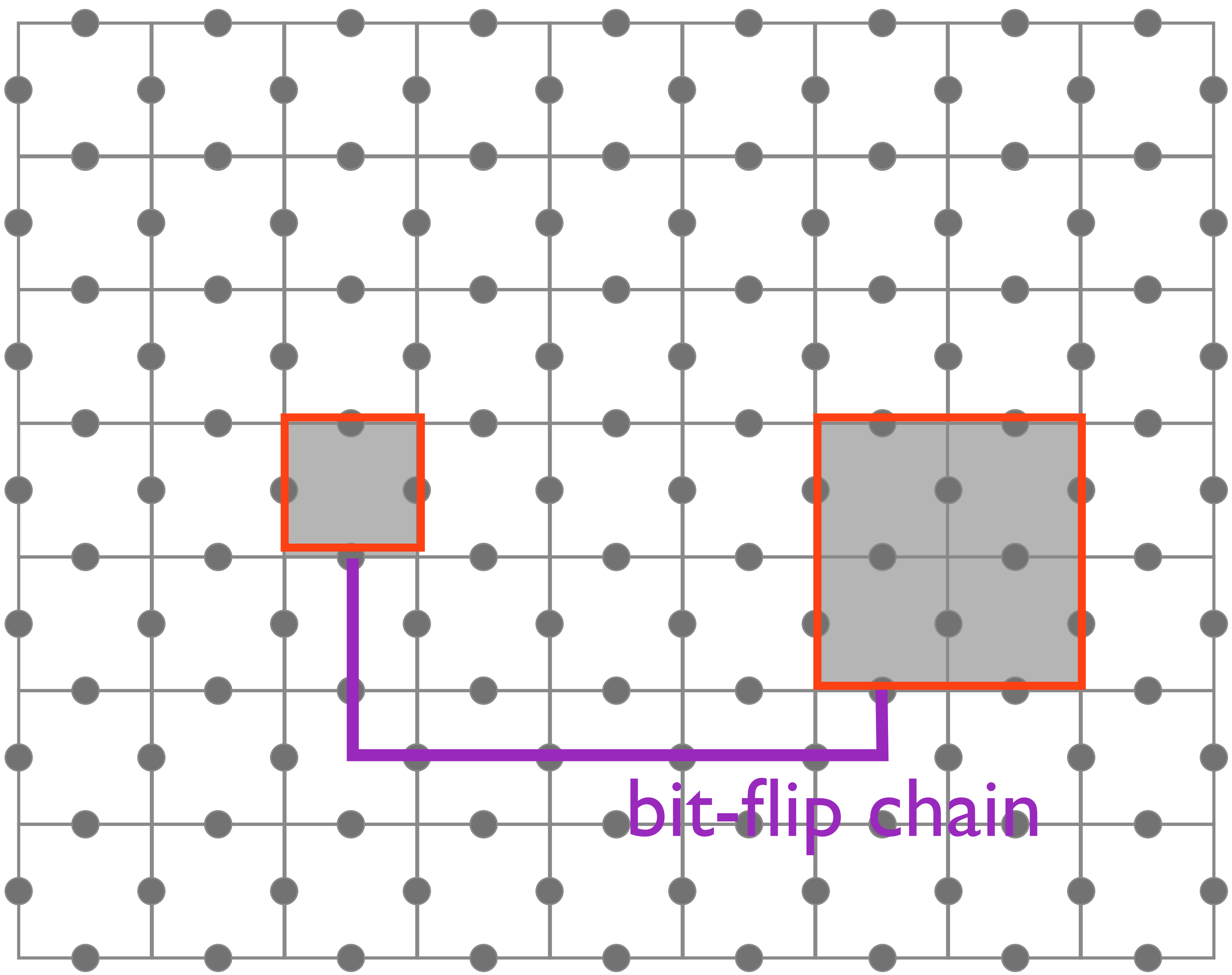
odd parity: 0

even parity: 1

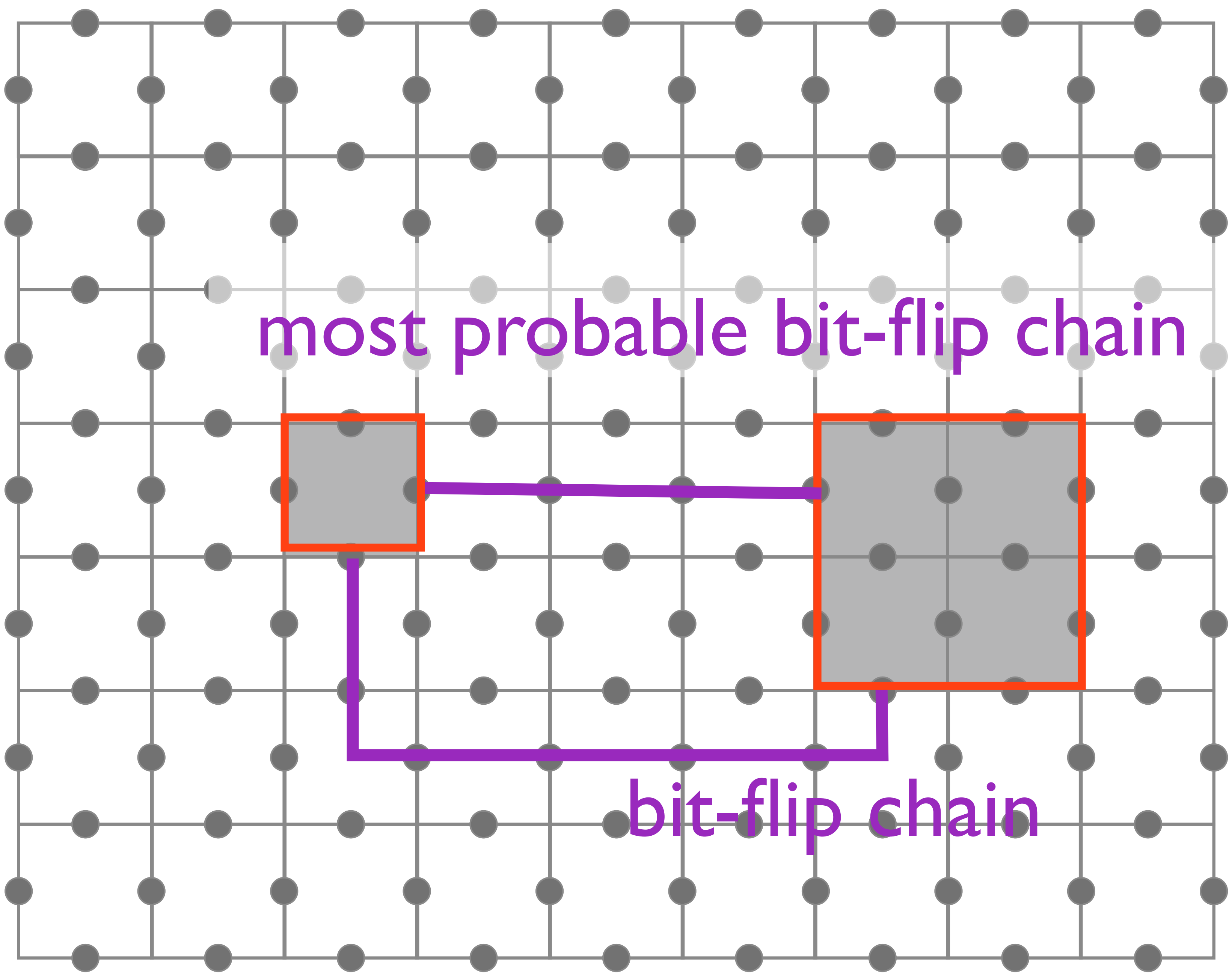
It's logical 1 here.

changed!

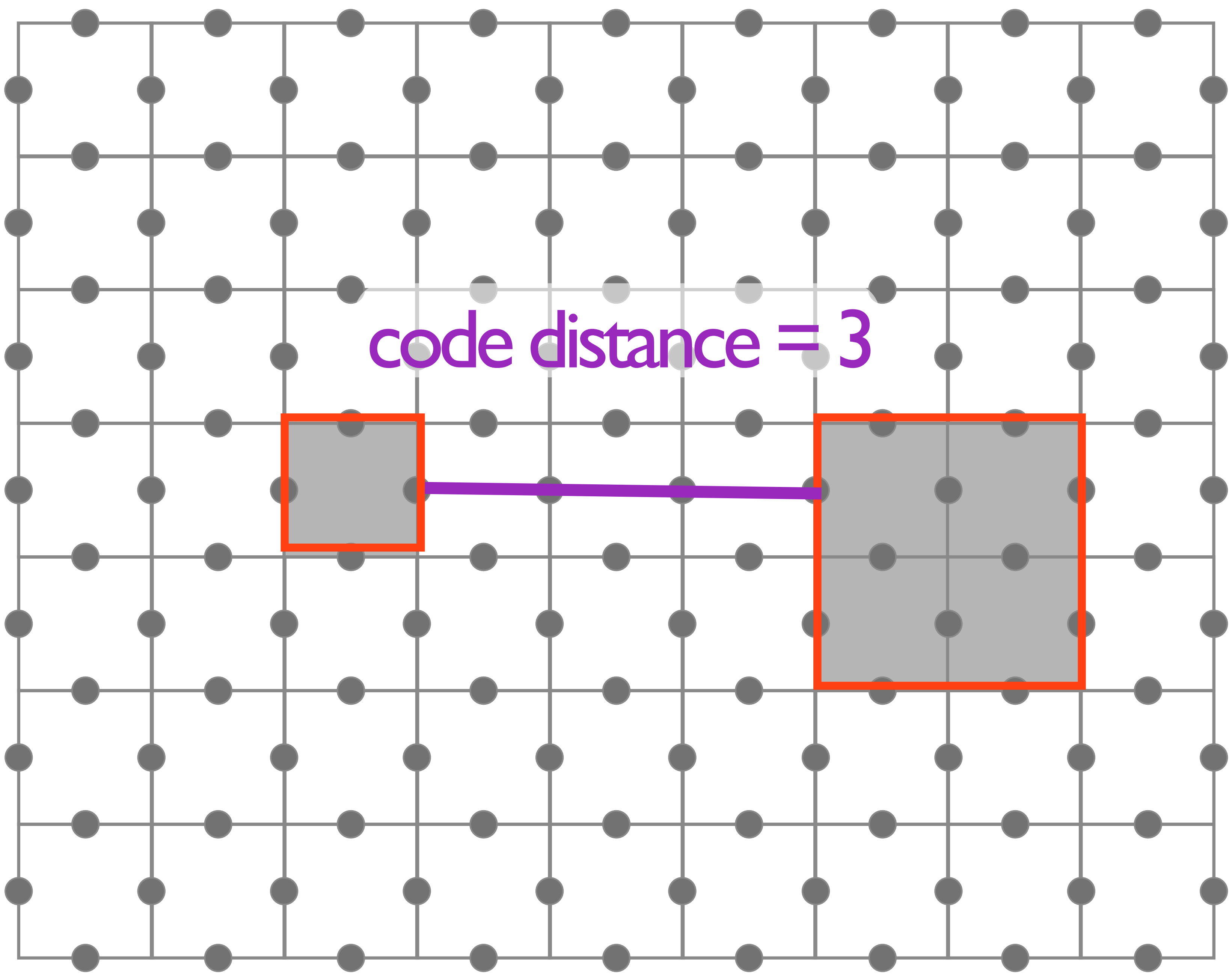
Surface Code
~Code distance~



Surface Code
~Code distance~

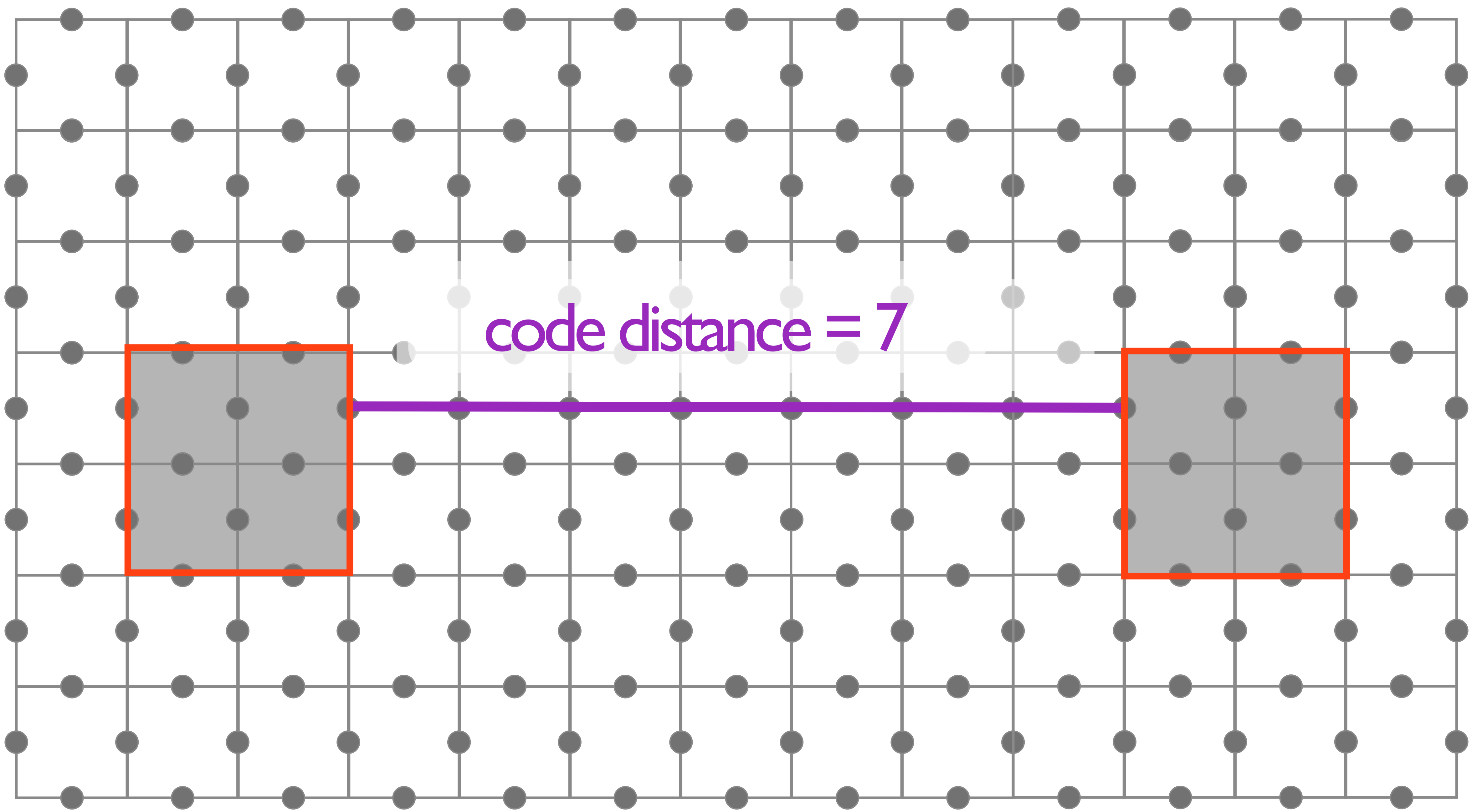


Surface Code
~Code distance~

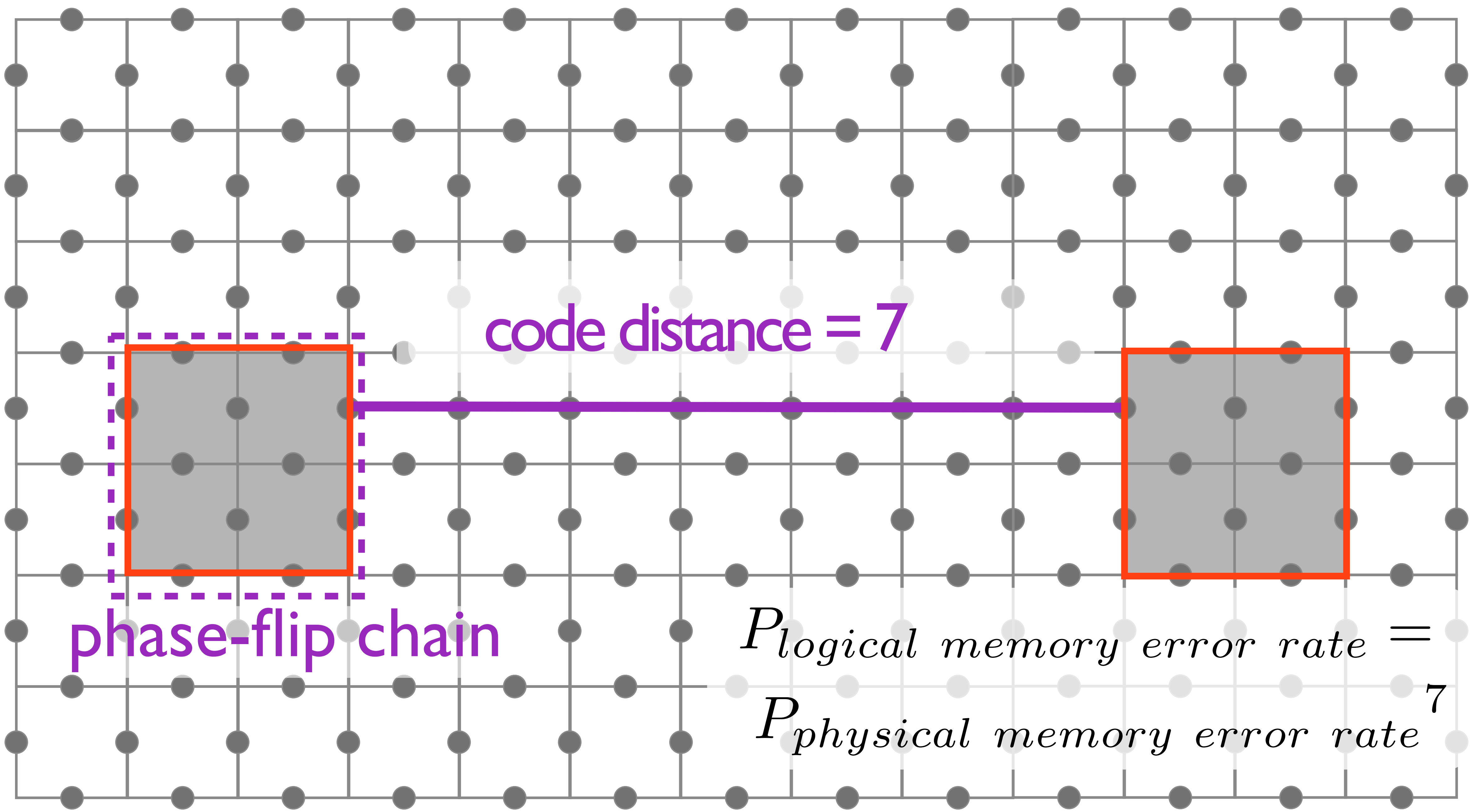


$$P_{logical\ memory\ error\ rate} = P_{physical\ memory\ error\ rate}{}^{4}$$

Surface Code
~Code distance~

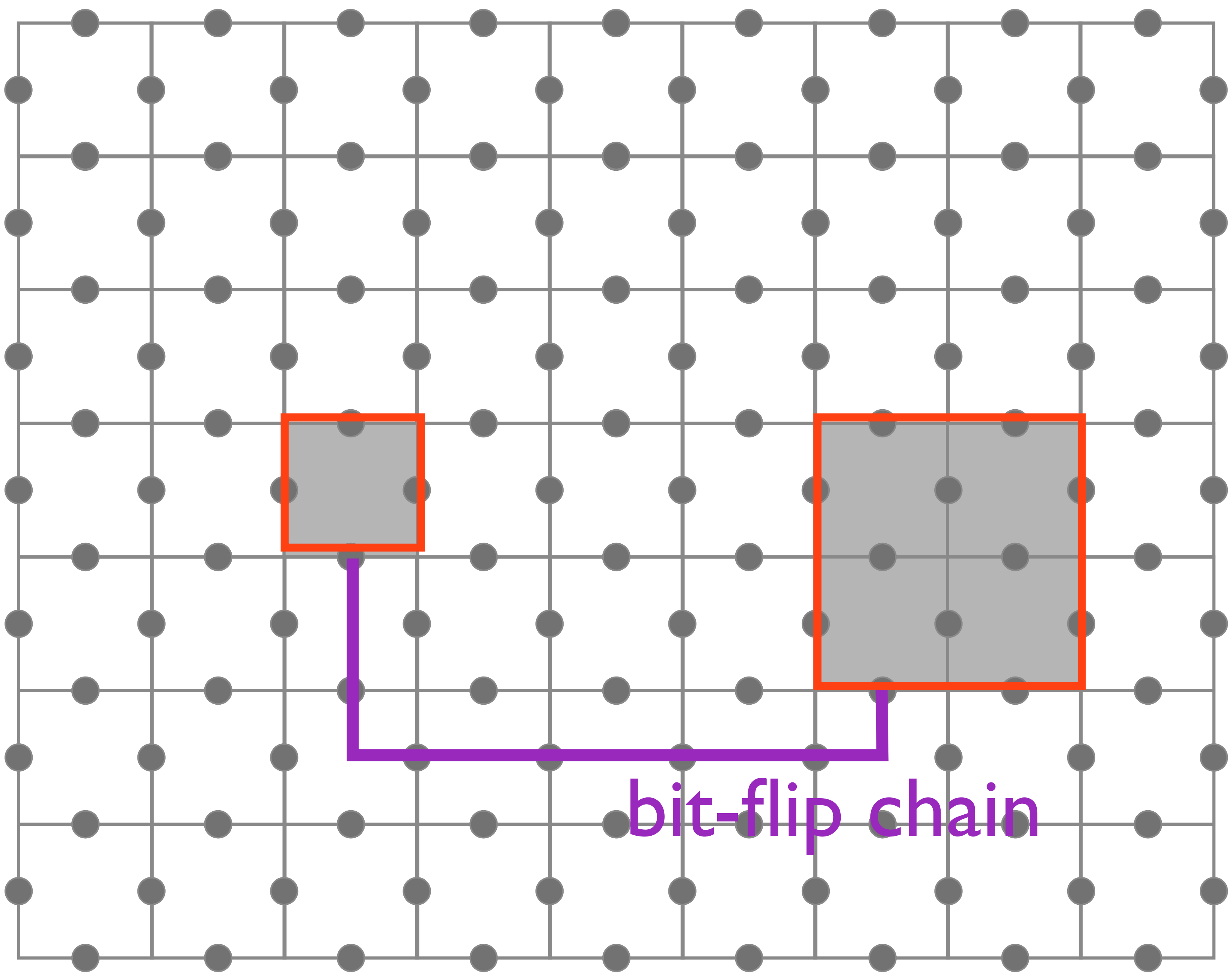


Surface Code
~Code distance~



$$P_{logical\ memory\ error\ rate} = P_{physical\ memory\ error\ rate}{}^{7}$$

Surface Code
~Code distance~



$$P_{logical\ gate\ error\ rate} = P_{physical\ gate\ error\ rate}{}^{7}$$

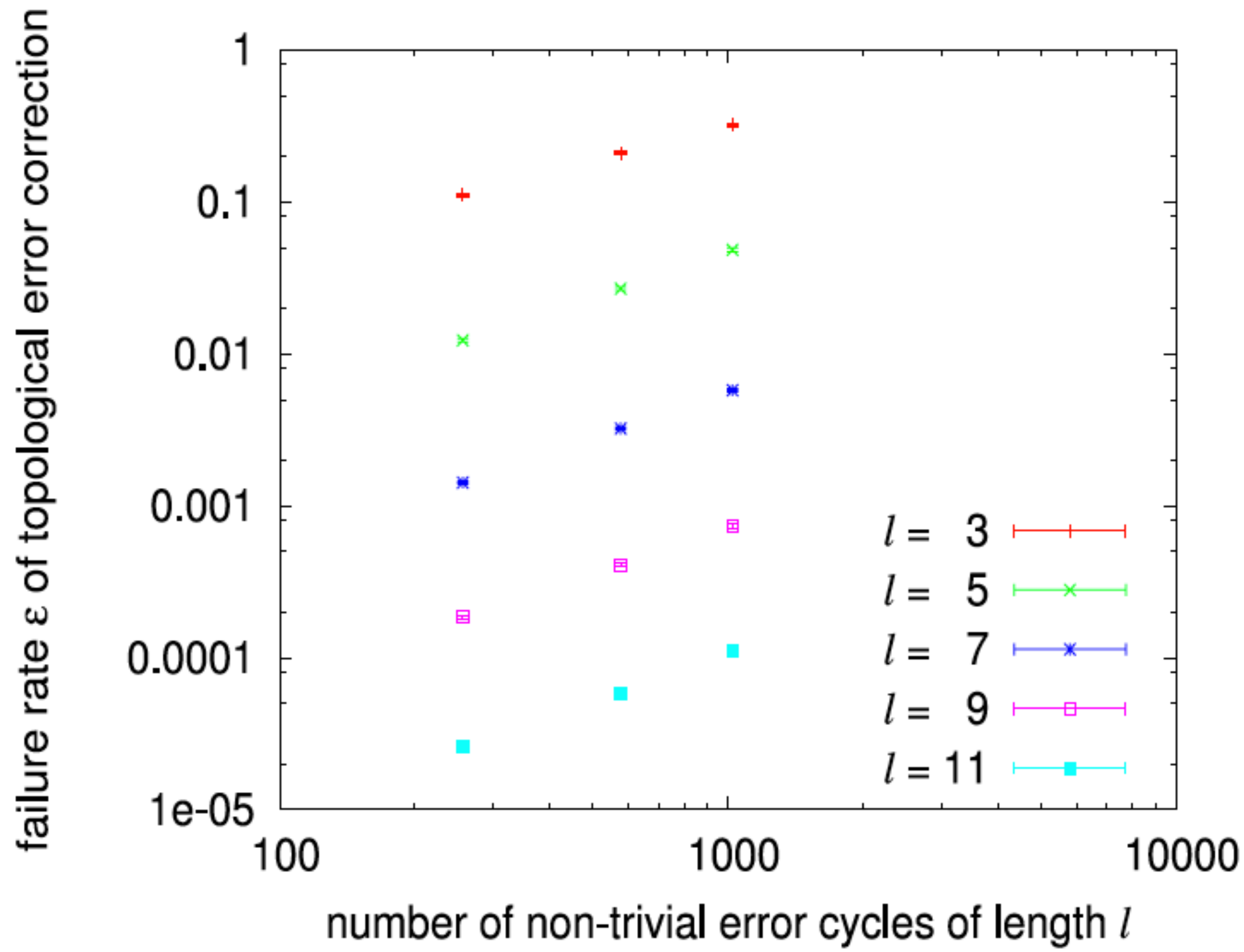
70

Exponential Suppression

- Classical simulation of small lattices
- Error rate

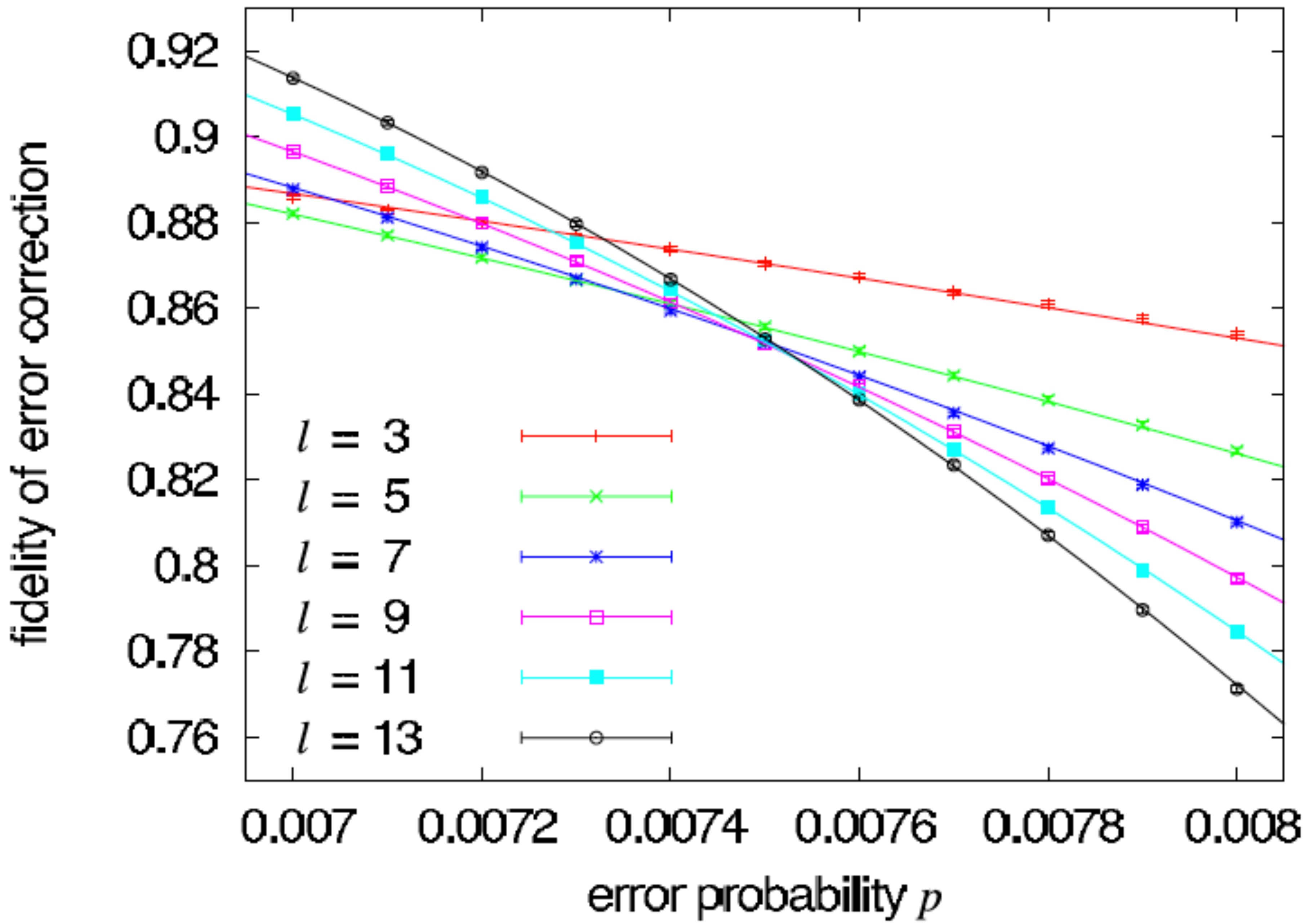
$$\epsilon_{\text{top}} \sim \exp(-\kappa(p)l)$$

- p = phy err rate
  K ~ 0.9
  l = min err chain length



- Still a polynomial correction needed
- Raussendorf et al., NJP 9, 2007

Threshold



- Gate = Memory = Meas errors = 0.75%
- Raussendorf et al., NJP 9, 2007

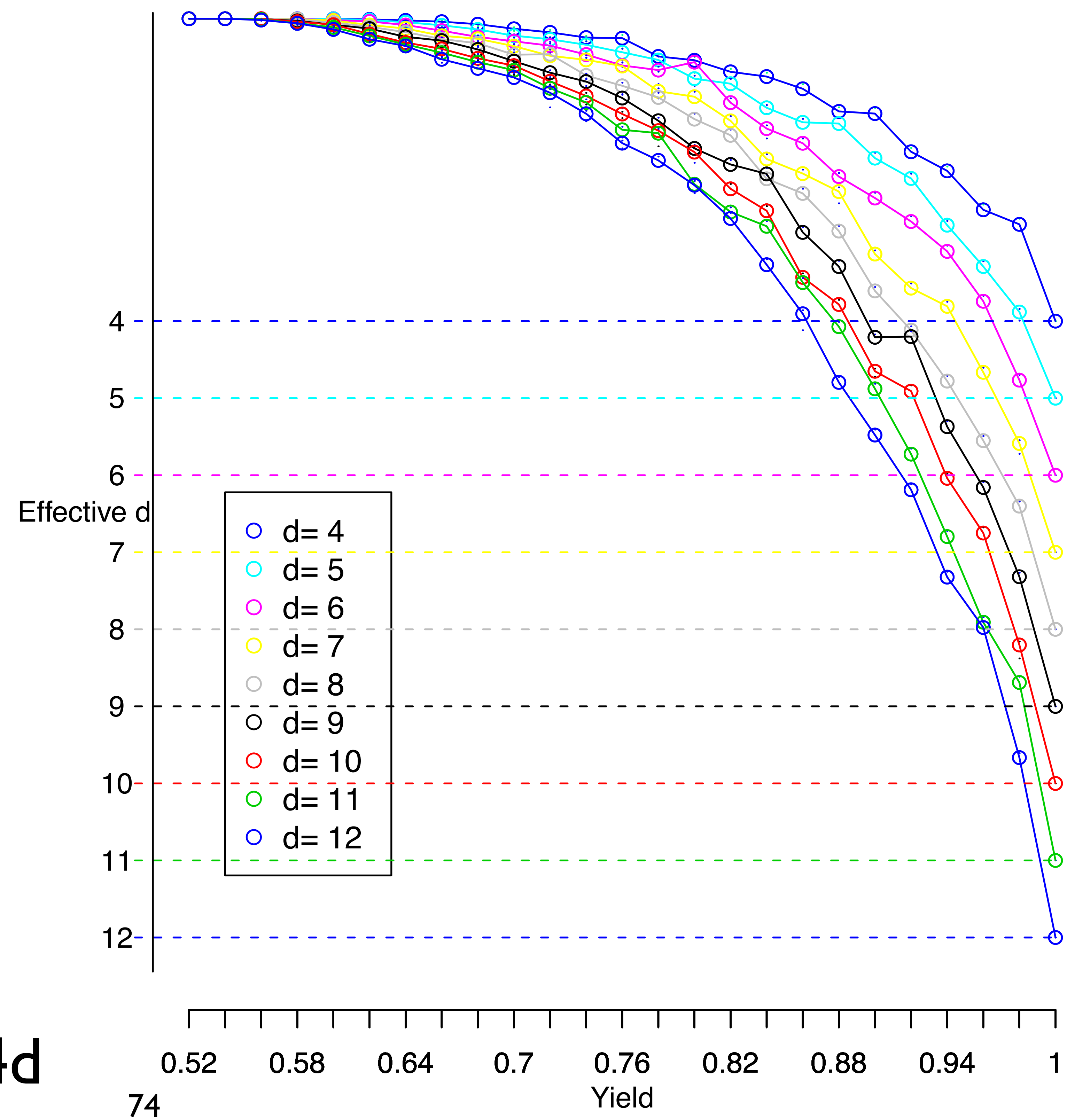
Advanced Topics

- 3-D version, talk to Simon
- Defective lattice
- Planar code
- Surface code communications
- Uses in distributed quantum computation

Defective lattice

Current estimate is that yield of 90% *halves* the effective code distance.

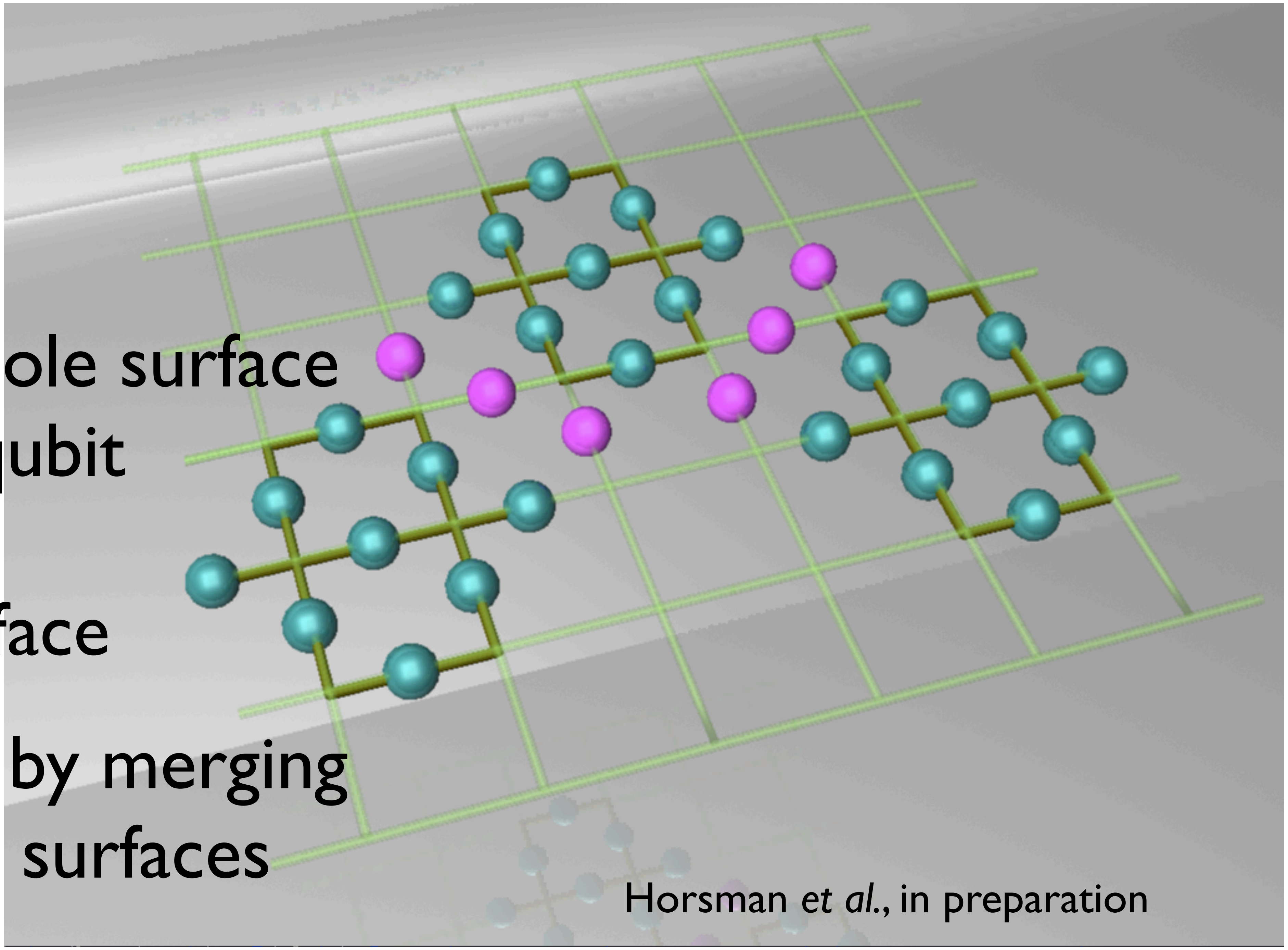
Nagayama *et al.*, in preparation



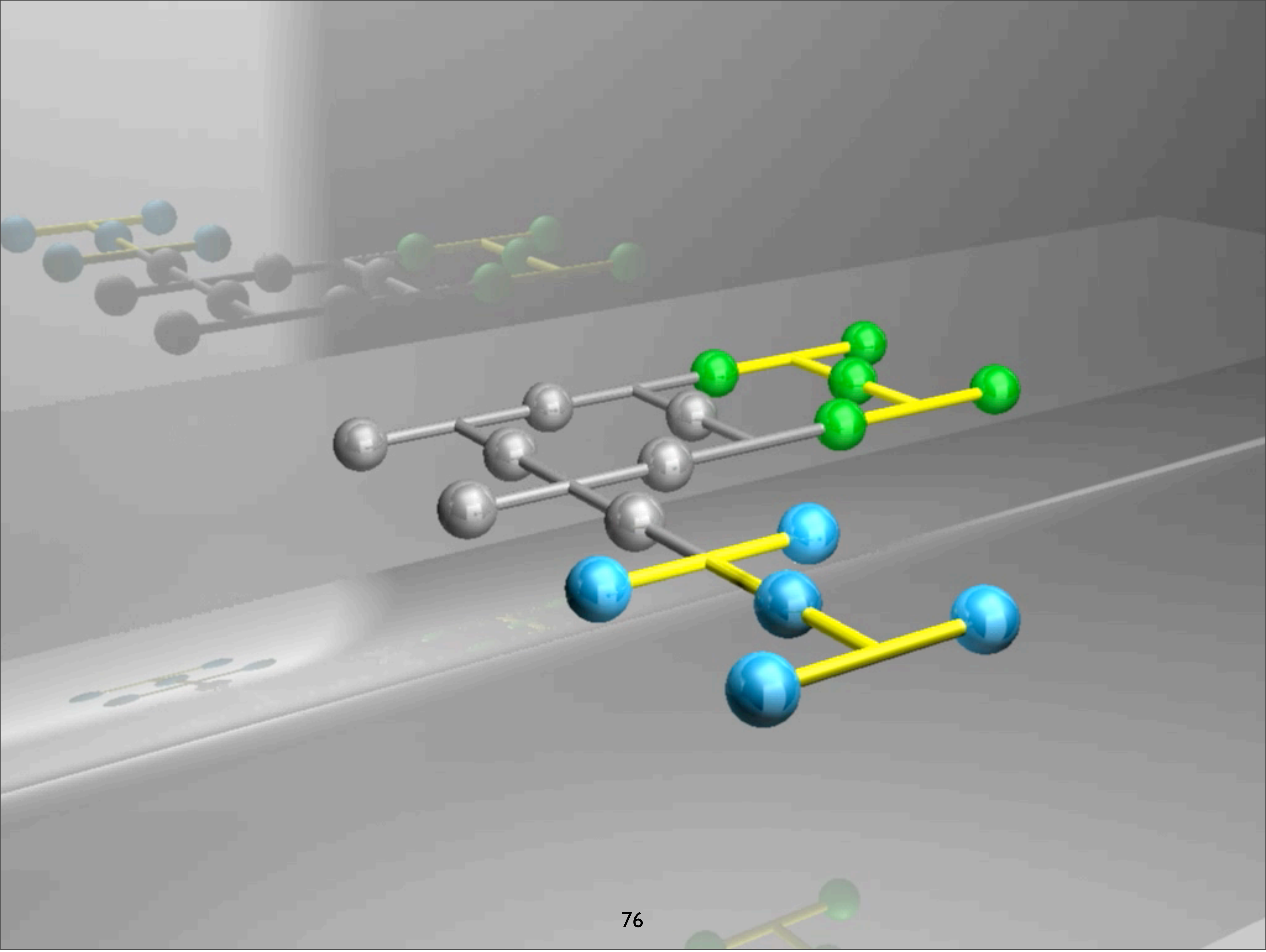
code distance = 4d

74

Planar Code



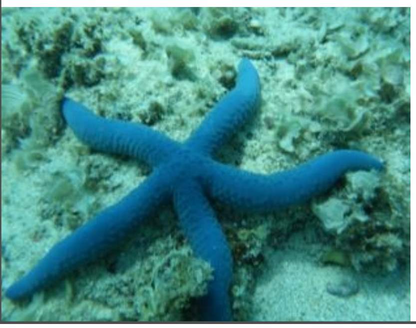
- Use one whole surface per logical qubit instead of holes in surface
- Gates done by merging and splitting surfaces
- Useful for small-scale experiments



Surface Code Strengths

Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)



Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)
- High threshold: 1.4% for gate, memory, measurement errors

Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)
- High threshold: 1.4% for gate, memory, measurement errors
- Flexible:
  - –strength of EC grows incrementally (compare to concatenated CSS codes)
  - –software-assigned resources
  - –easy movement of logical qubits

Surface Code Strengths

- Simple, 2-D or 3-D nearest-neighbor-only operation (physical feasibility high!)
- High threshold: 1.4% for gate, memory, measurement errors
- Flexible:
  - –strength of EC grows incrementally (compare to concatenated CSS codes)
  - –software-assigned resources
  - –easy movement of logical qubits
- Supporting classical processing achievable

Key References

- clearest explanation:
  Fowler *et al.*, PRA 80, 052312 (2009)
  (source of many of the figures)
- detailed paper:
  Raussendorf *et al.*, NJP 9, 199 (2007)
- cryptic seminal paper:
  Raussendorf and Harrington, PRL 98, 190504 (2007)
- Surface code communication:
  Fowler *et al.*, PRL 104, 180503 (2010)
- Defects in the surface code:
  Stace *et al.*, PRL 102, 200501 (2009)

8 Hours of Lecture on Surface Code

- http://aqua.sfc.wide.ad.jp/Publications.html

Preview: What kind of system can run surface code effectively?

- Billions of qubits
- GHz physical gates
- Millisecond memory lifetimes
- Error rate ~0.1%
- ~1 year to factor 2048-bit number
- Stay tuned...