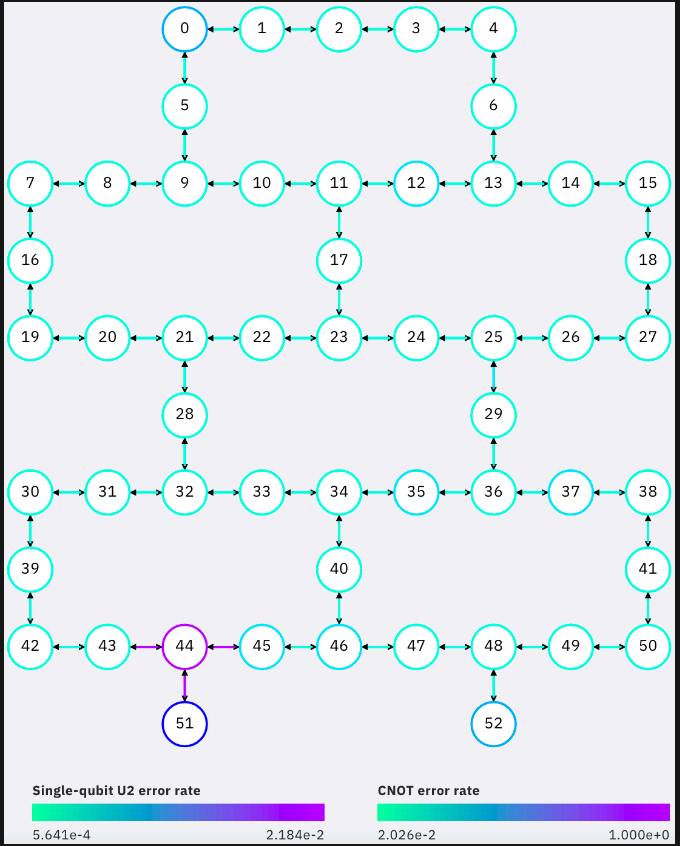
Benchmarking with Quantum Error Correction

James Wootton
IBM Quantum, IBM Research - Zurich

IBM **Quantum**

How good is a quantum device?

- We often use fidelities, T1 and T2 etc to characterize a device
- It doesn't tell the whole story
  - What is the form of the noise?
  - How is it transformed by gates?
  - How does it affect our ability to run algorithms?
- Quantum volume aims to capture this
- It tells us how well the device can run the QV circuit

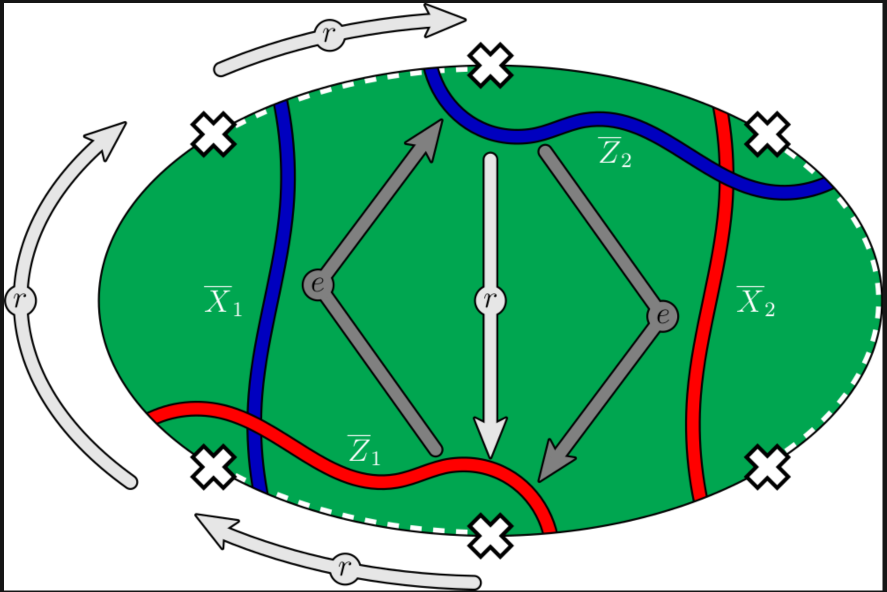


How good is a quantum device?

- Many different ways to design an algorithm in the near-term
  - Will respond differently to noise
  - Have different compatibility with error mitigation
- In the FT era, things should simplify
  - Everything built on top of QEC

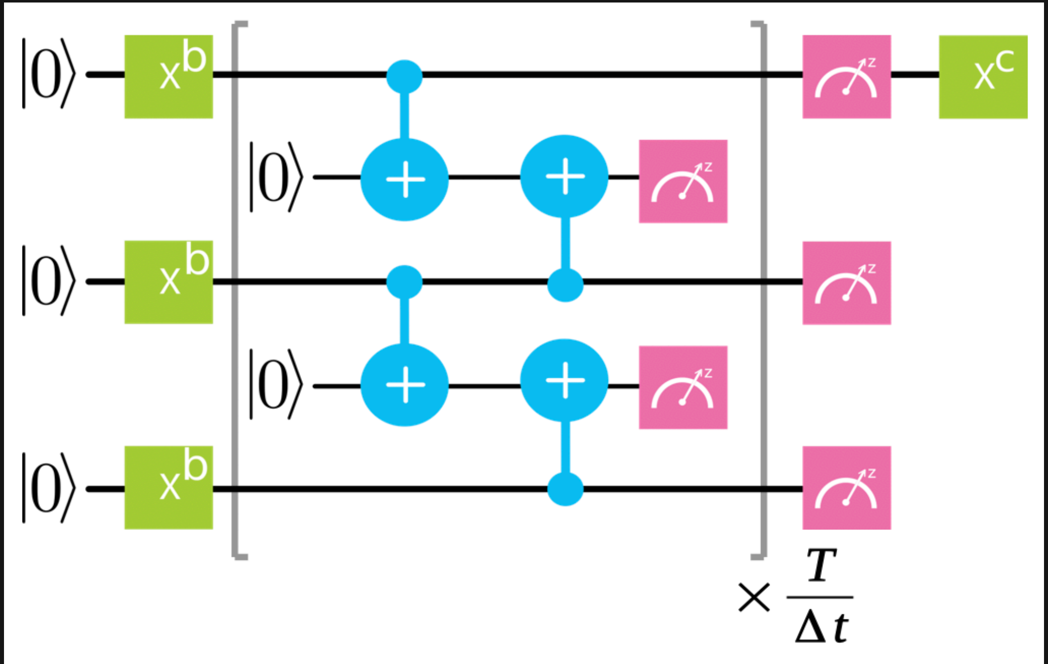
- An FTQC is basically a QEC machine
- Algorithms are minor perturbations

- So how well can a device do QEC?



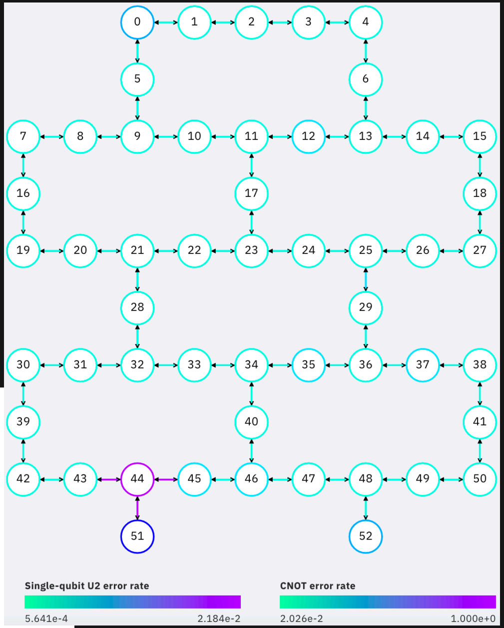
How good is QEC?

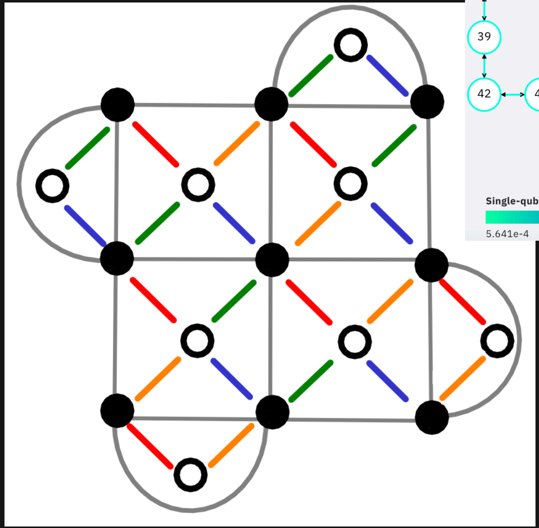
- Basic aim is to get a long lifetime
- Increasing this requires
  - Increasing qubit number
  - Maintaining good connectivity
  - Increasing circuit depth
  - Selective measurements throughout
  - Maintaining low noise
- In short, everything we need for QC!



How good is QEC today?

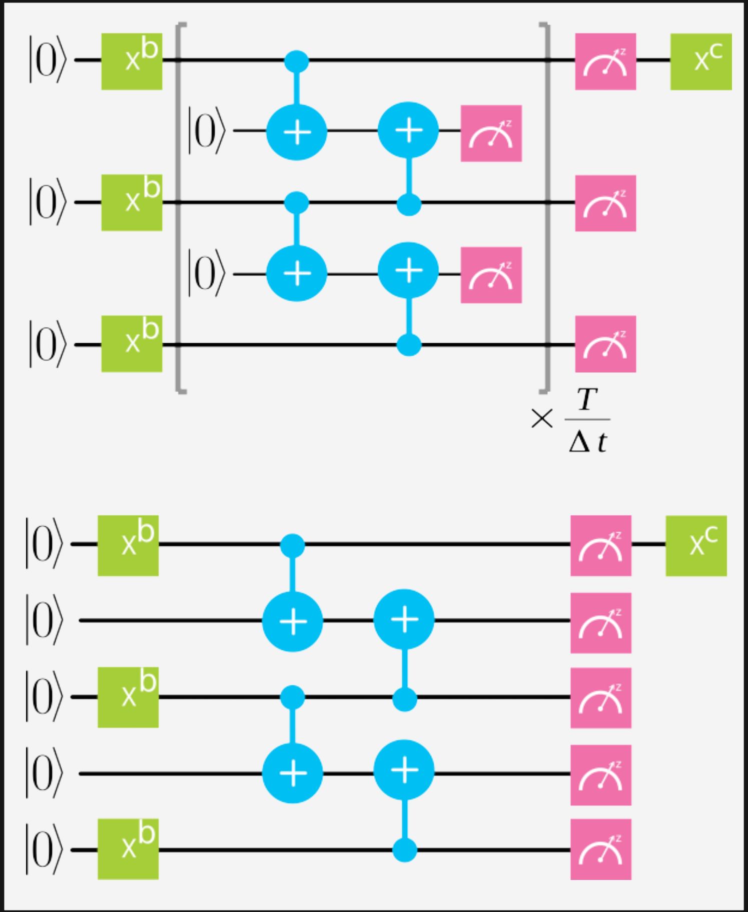
- Currently we have
  - Up to 53 qubits
  - Heavy hexagonal connectivity
  - All measurements at end
- Not even enough for a minimal surface code!





How good is QEC today?

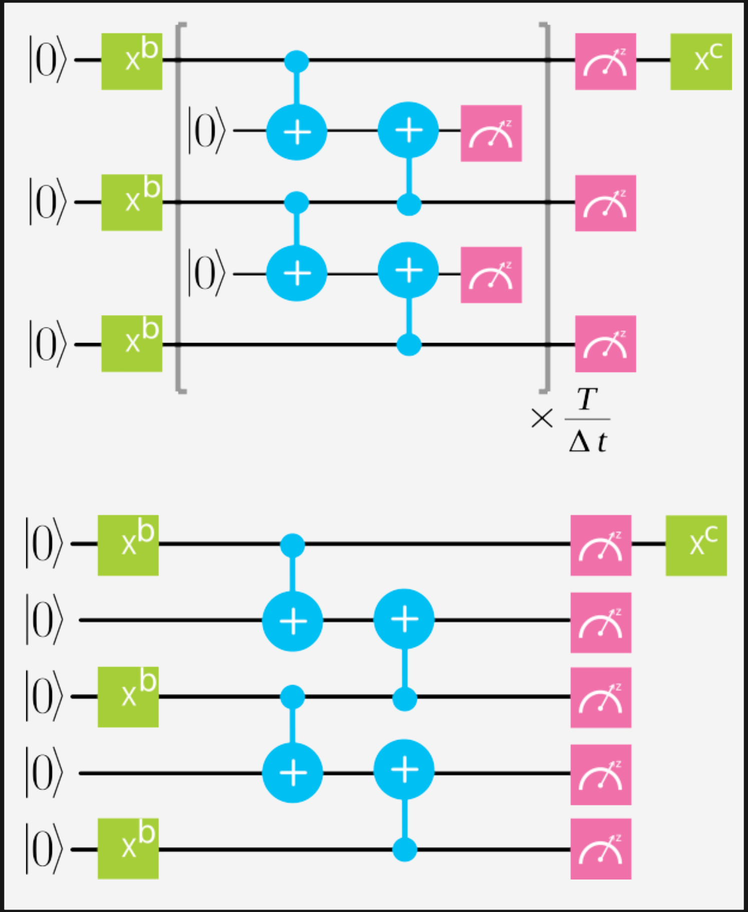
- We need to at least test the standard methodology
  - Encode bit values
  - Detect errors using stabilizer measurements
  - Correct during decoding
- Simplest way is using the repetition code
  - Needs at least 5 qubits
  - Very flexible on connectivity
  - Detects and corrects bit flips



Repetition code experiment

- For *d* repetitions, we need
  - *d* code qubits
  - *d-1* ancillae
- We implement a single round
  - Encode 0 or 1
  - Measure standard syndromes (error detection)
  - Measure code qubits (error detection and readout)
  - Use output to infer input

- How often is the output correct?

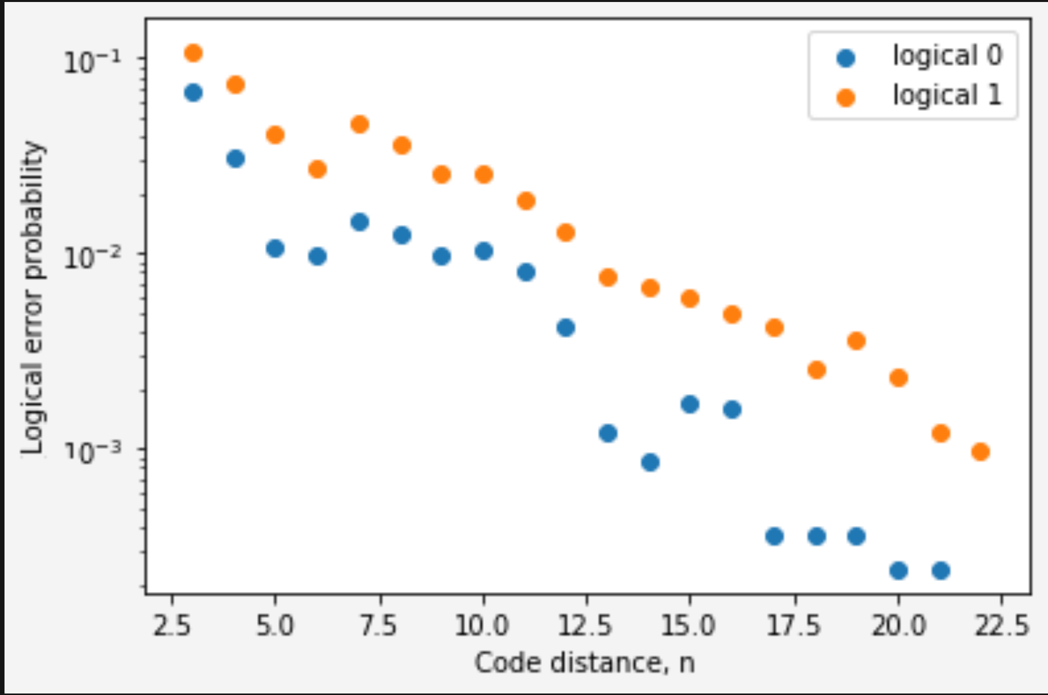


Repetition code experiment

– Codes of up to *d=5* done with multiple rounds

J. Kelly, et al., Nature 519, 66–69 (2015)

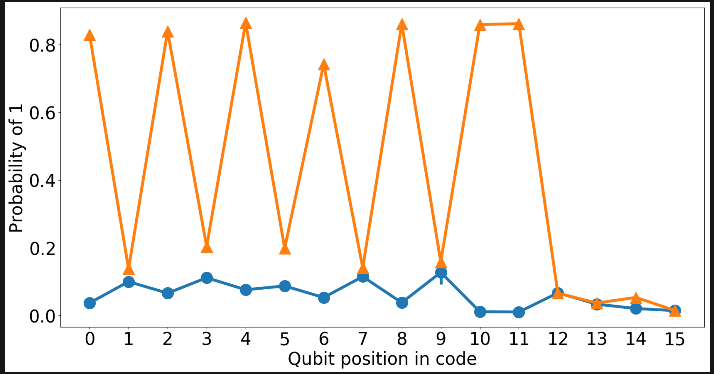
– We can probe the other extreme: increasing *d*



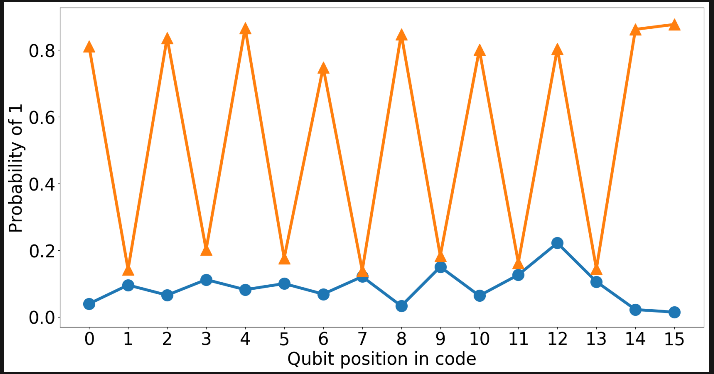
16 qubit experiments

- First experiments done with *ibmqx3/rueschlikon*
  - Wootton and Loss, Phys. Rev. A 97, 052313 (2018)
  - Naveh, et al., Proceedings of the 2018 Design, Automation & Test in Europe(DATE)(2018).
- From final states we see
  - Fidelities of 80-90%
  - Extending a code doesn't affect existing parts

d=6



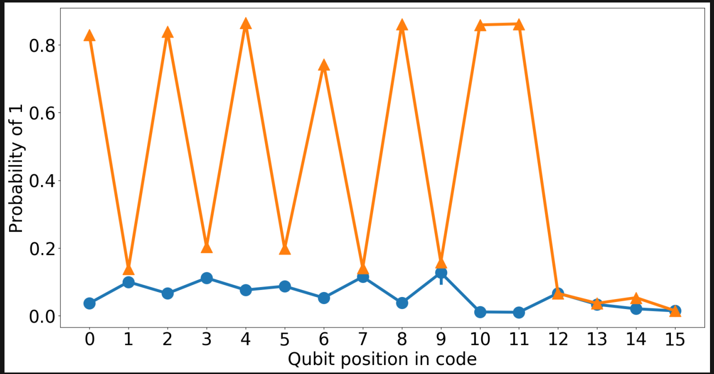
d=8



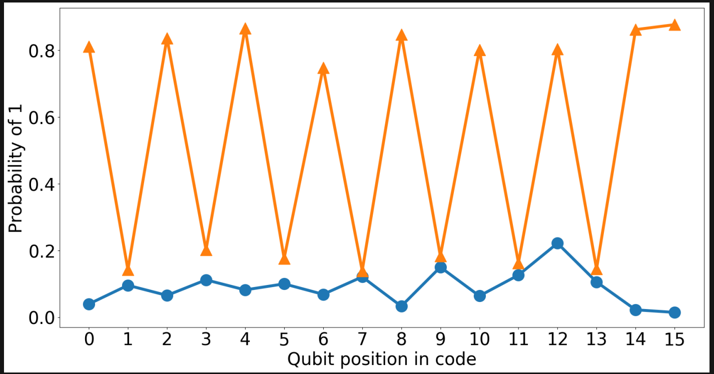
16 qubit experiments

- First experiments done with *ibmqx3*
  - Wootton and Loss, Phys. Rev. A 97, 052313 (2018)
  - Naveh, et al., Proceedings of the 2018 Design, Automation & Test in Europe(DATE)(2018).
- From final states we see
  - Fidelities of 80-90%
  - Extending a code doesn't affect existing parts

d=6



d=8

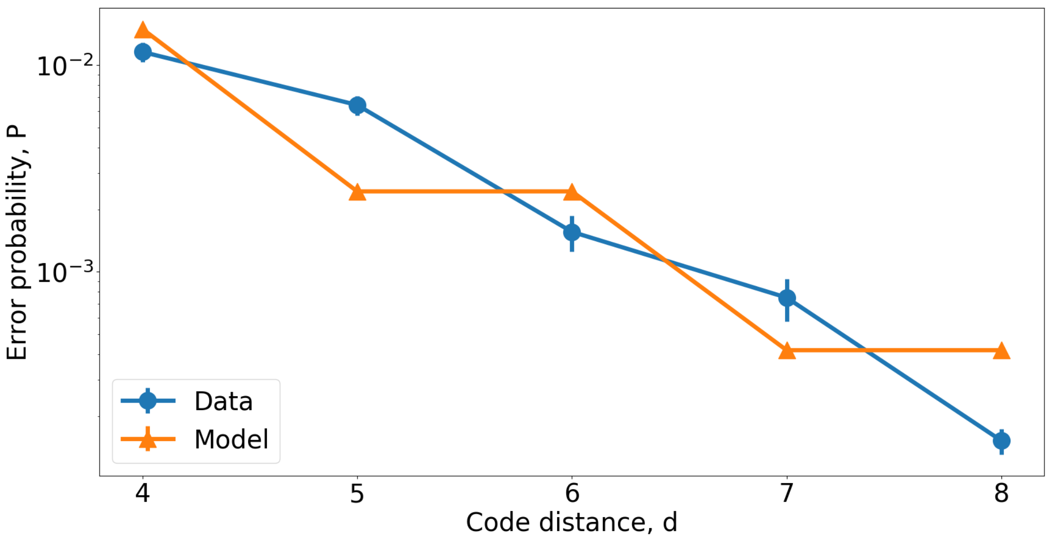


16 qubit experiments

- For logical error rate, we expect an exponential decrease
- A simple, single parameter model of this is

$$\left(\frac{p}{1-p}\right)^{\lceil d/2 \rceil}$$

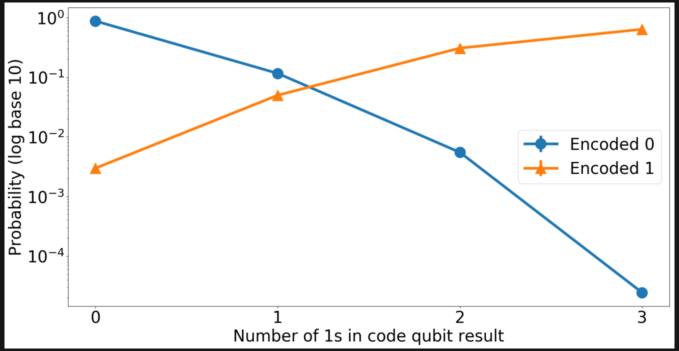
- Notice than even/odd effect is expected
- Exponential decrease is observed
- Even/odd effect is inverted!



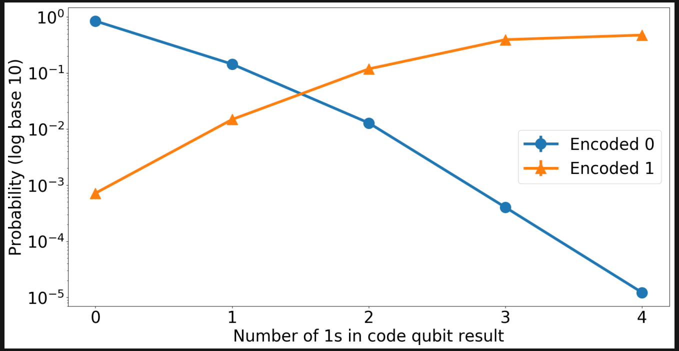
16 qubit experiments

- Lookup table is used for decoding
- Many instances run to see probabilities for outcomes for each encoded value
- Reversing this we get most likely encoded value given an outcome
- This differs strongly from majority voting

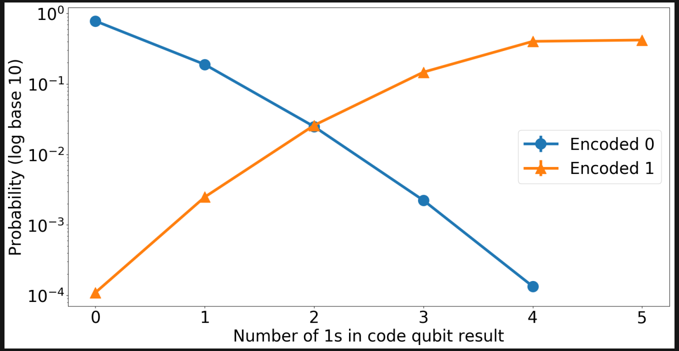
d=3



d=4



d=5

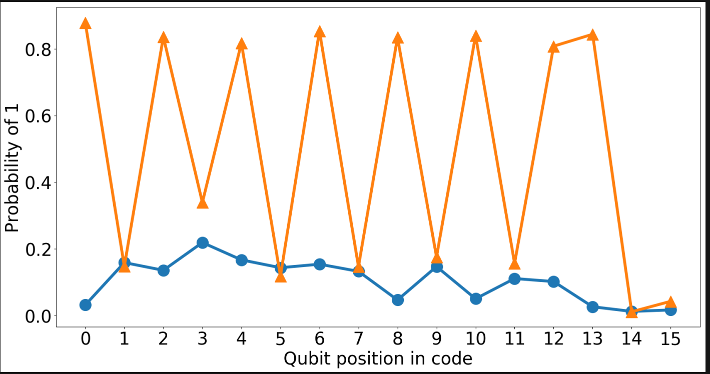


16 qubit experiments

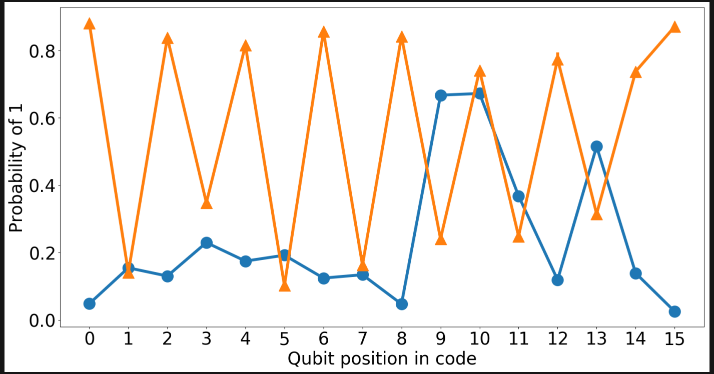
IBM **Quantum**

- Experiment repeated on *ibmqx4*
- Results were not good: no exponential decay
- Adding qubits caused strong non-local effects
- We cannot take success for granted!

d=7



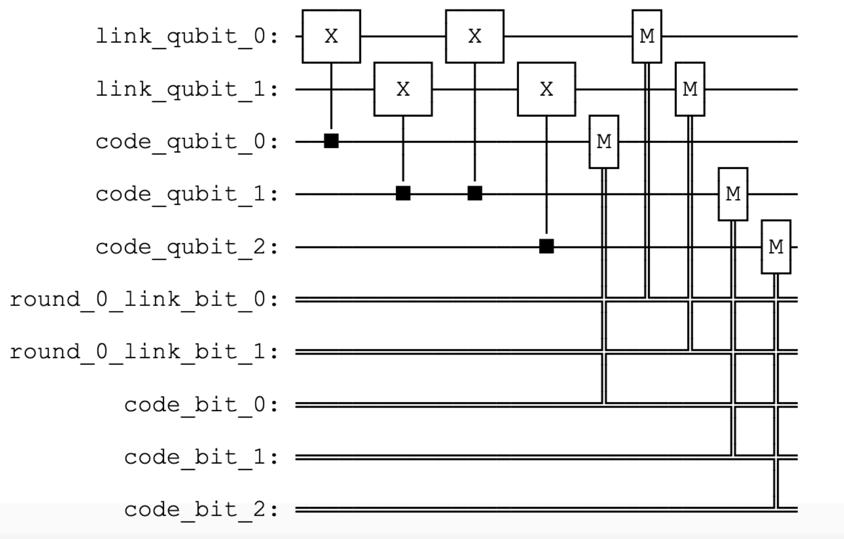
d=8



`topological_codes` in Qiskit Ignis

- Previous experiments used custom code
- Now let’s make something resuable
- Module in Qiskit Ignis for QEC benchmarking experiments
- Currently has
  - 1 code (repetition)
  - 1 decoder (MWPM)
- But more are on the way

```
d = 3
T = 1
code = RepetitionCode(d,T)

code.circuit['0'].draw()
```



`topological_codes` in Qiskit Ignis

- Two circuits per code that need to be run
- Required output is the counts dictionary
- The bit strings are transformed into a more QEC friendly form

```
circuits = code.get_circuit_list()
job = execute( circuits, Aer.get_backend('qasm_simulator') )
raw_results = {}
for log in ['0','1']:
    raw_results[log] = job.result().get_counts(log)

raw_results
```

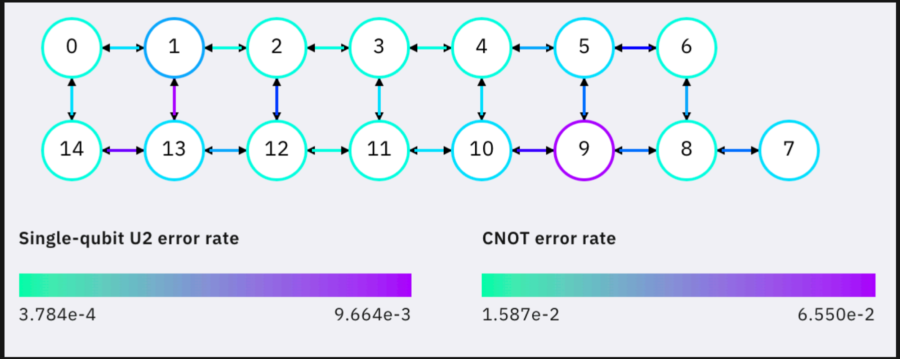
```
{'0': {'000 00': 1024}, '1': {'111 00': 1024}}
```

```
results = code.process_results( raw_results )

results
```

```
{'0': {'0 0  00 00': 1024}, '1': {'1 1  00 00': 1024}}
```

`topological_codes` in Qiskit Ignis

- Important to know which qubits are being used
- Remapping also remaps noise, making uncorrectable errors

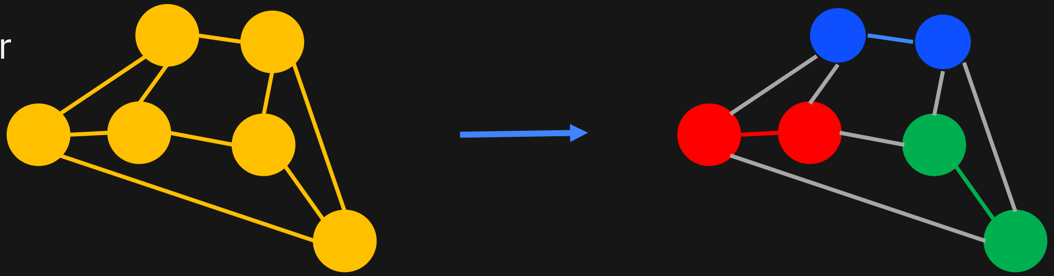



```python
line = [13,14,0,1,2,12,11,3,4,10,9,5,6,8,7]

initial_layout = {}
for j in range(d):
    initial_layout[code.code_qubit[j]] = line[2*j]
for j in range(d-1):
    initial_layout[code.link_qubit[j]] = line[2*j+1]

initial_layout
```

```
{Qubit(QuantumRegister(3, 'code_qubit'), 0): 13,
 Qubit(QuantumRegister(3, 'code_qubit'), 1): 0,
 Qubit(QuantumRegister(3, 'code_qubit'), 2): 2,
 Qubit(QuantumRegister(2, 'link_qubit'), 0): 14,
 Qubit(QuantumRegister(2, 'link_qubit'), 1): 1}
```

`topological_codes` in Qiskit Ignis

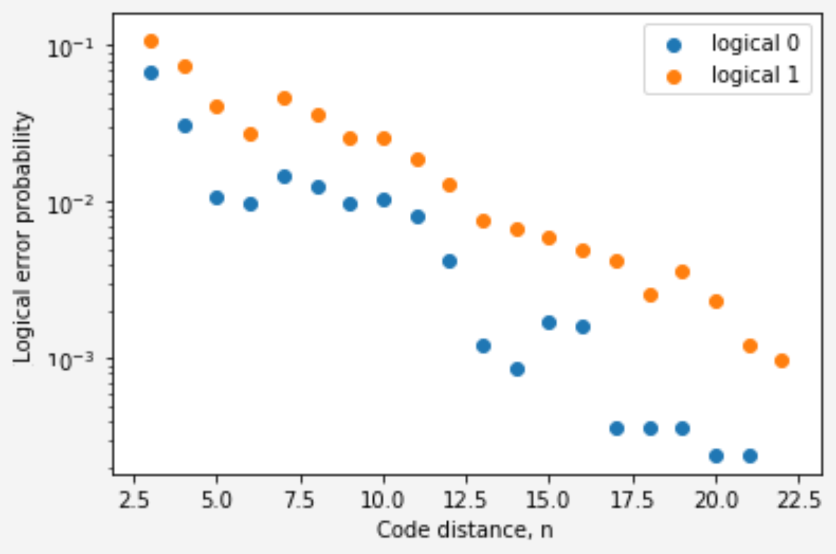
- Decoding done through graph theoretic analysis
- Artificial Pauli ‘errors’ added into circuit
- Graph created based on how the output changes
- Any given output corresponds to subgraph
- Analysis used to determine most likely error

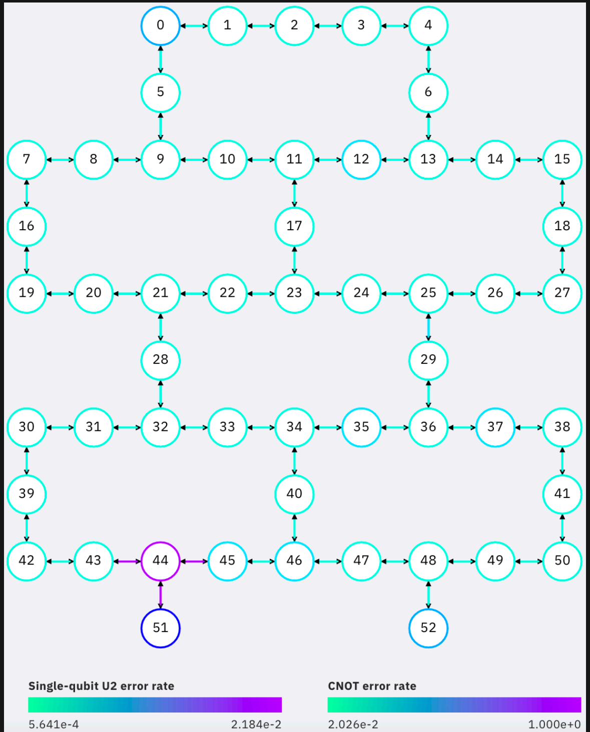


```
decoder = GraphDecoder(code)

graph = decoder.make_error_graph('0 1  000 000 001')
```

43 qubit results

- James R. Wootton 2020 Quantum Sci. Technol. doi.org/10.1088/2058-9565/aba038
- Run on *rochester* for d=3 to d=23 codes (up to 43 qubits)
- Results consistent with exponential decay



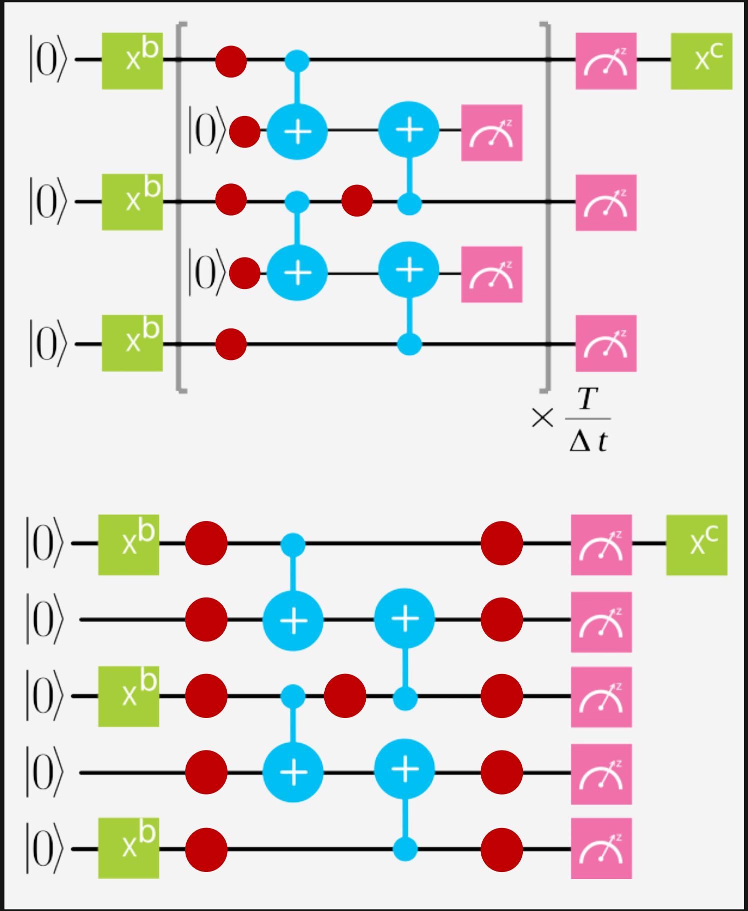


43 qubit results

- The results can be used to estimate probabilities of each error
  - Paulis at each point in circuit
- Not really comparable to anything, but let's compare to readout error probs

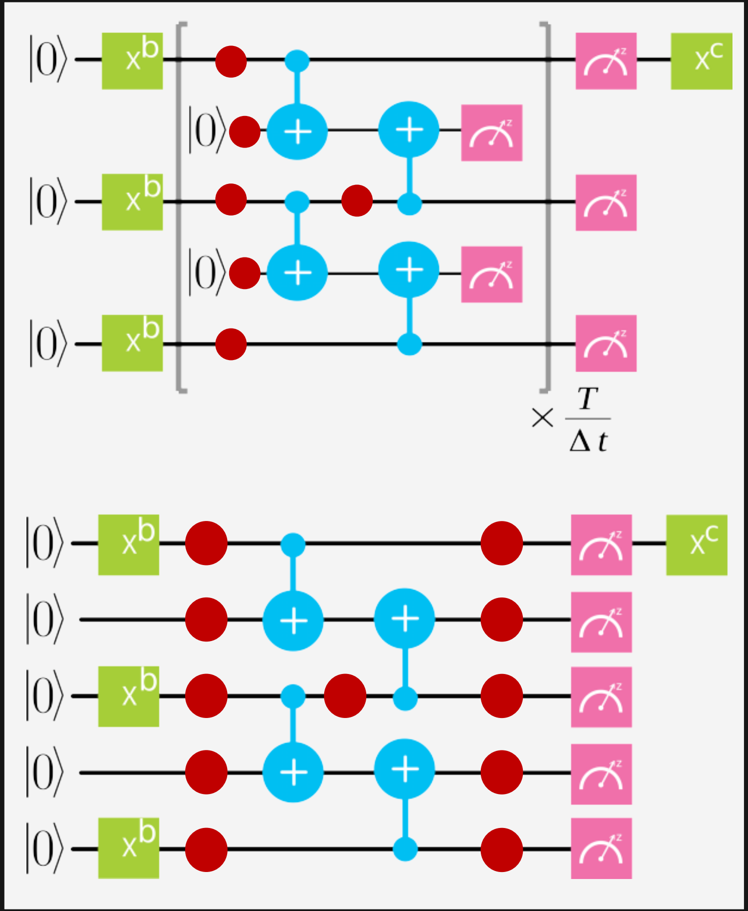
```
count: 85.0
mean: 0.15019428099809715
std: 0.11236995205935389
min: 0.026545002073828285
25%: 0.05599450360700791
50%: 0.11155859846959323
75%: 0.20286006128702758
max: 0.41629711751662973
```

```
count: 127.0
mean: 0.06850161154431056
std: 0.05891171361248407
min: 0.019374999999999992
25%: 0.03708774898932393
50%: 0.046021125148897085
75%: 0.0857536432141715
max: 0.33624999999999994
```



Future experiments

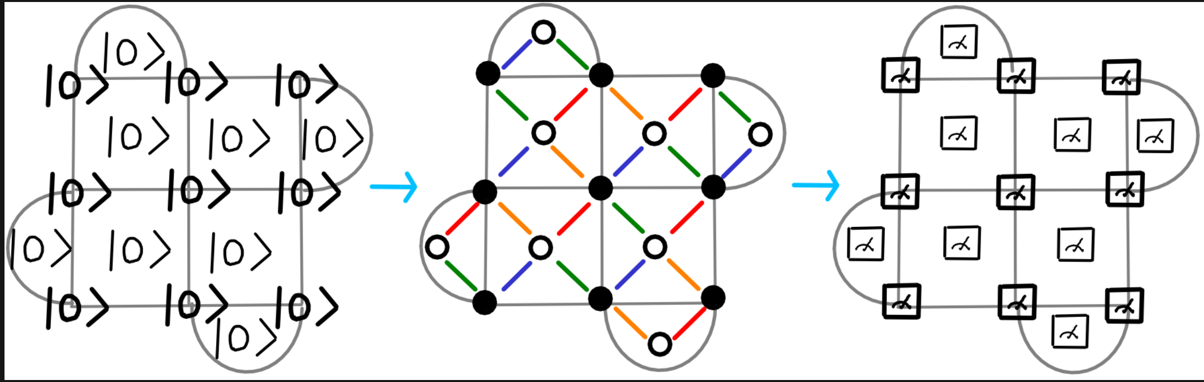
- We can, of course, make even bigger repetition codes
- For T>1, we can investigate time dependence of errors
  - Calculate probabilities of errors for different (qubit, depth)
  - How do these vary over the course of the circuit?



Future experiments

IBM **Quantum**

– We also want a proper quantum code, such as *d=3* surface code



– Might take a while, but we should try to be ready!

Thanks for your attention!