INTRO TO QUANTUM COMPUTING

Week 22 Lab

RANDOMIZED BENCHMARKING

Corbin McElhanney

April 20, 2021

© 2021 The Coding School

PROGRAM FOR TODAY

- Canvas attendance quiz
- Pre-lab zoom feedback
- Lab content
- Post-lab zoom feedback

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

CANVAS ATTENDANCE QUIZ

- Please log into Canvas and answer your lab section's quiz

Lab Number: 1 | Quiz Password: 8437

- **Question:** Do you agree or disagree with the following statement? This course provides enough resources to answer my questions about the content. (e.g. Piazza, Office Hours, Homework Review Sessions, etc.)
- [Optional] What additional resources would be helpful?

- **This quiz is not graded, but counts for your lab attendance!**

© 2021 The Coding School

PRE-LAB ZOOM FEEDBACK

On a scale of 1 to 5, how would you rate your understanding of this week's content?

- 1 –Did not understand anything
- 2 – Understood some parts
- 3 – Understood most of the content
- 4 – Understood all of the content
- 5 – The content was easy for me/I already knew all of the content

In lecture this week, Amir described criteria and benchmarks for physical qubit systems

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

LEARNING OBJECTIVES FOR LAB 22

Implementing randomized benchmarking

- Errors in quantum computing
- Characterizing errors through benchmarking
- The steps in randomized benchmarking
- Coding randomized benchmarking

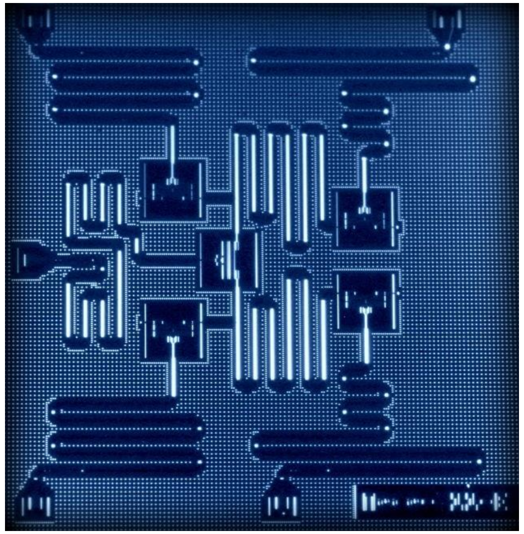
© 2021 The Coding School
Canvas attendance quiz passcode: 8437

PHYSICAL QC SYSTEMS

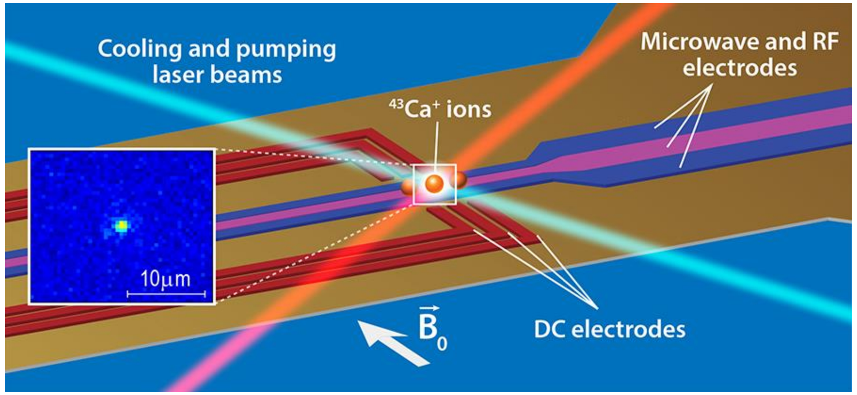
- There are several different systems being developed for qubits

- **Why?** Right now it is not clear which one will have the best performance for different types of applications

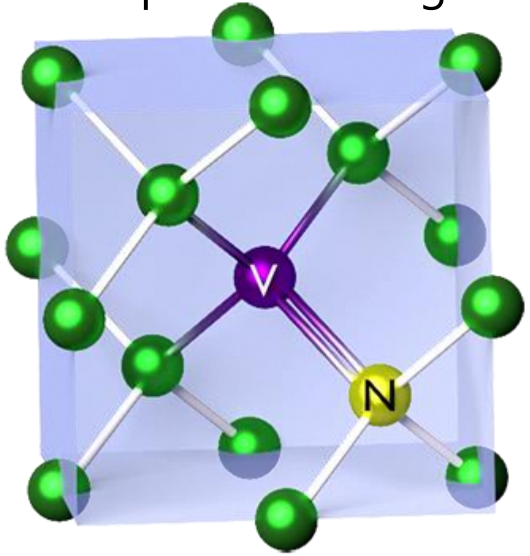
- Key performance metric - **errors**



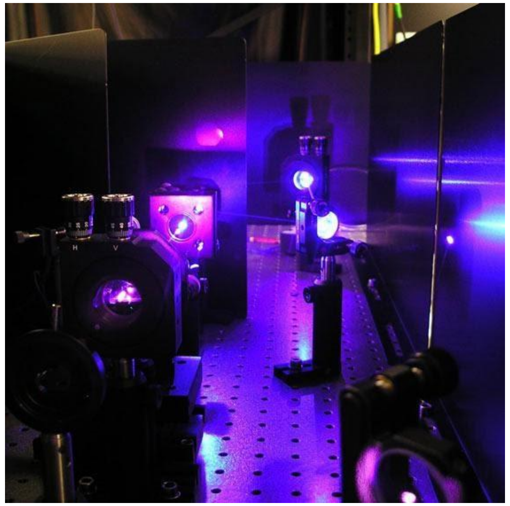
superconducting



trapped ion



diamond NV centers



photonic

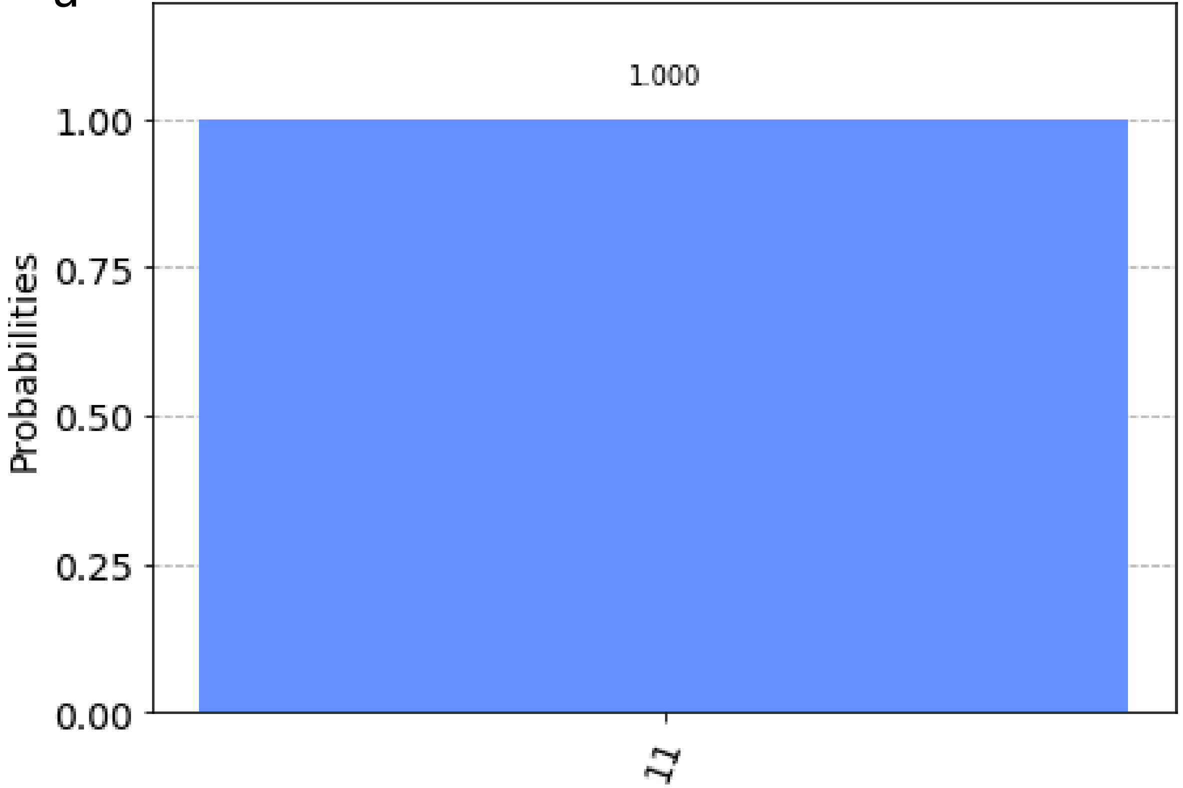
© 2021 The Coding School
Canvas attendance quiz passcode: 8437

ERRORS IN QUANTUM COMPUTING

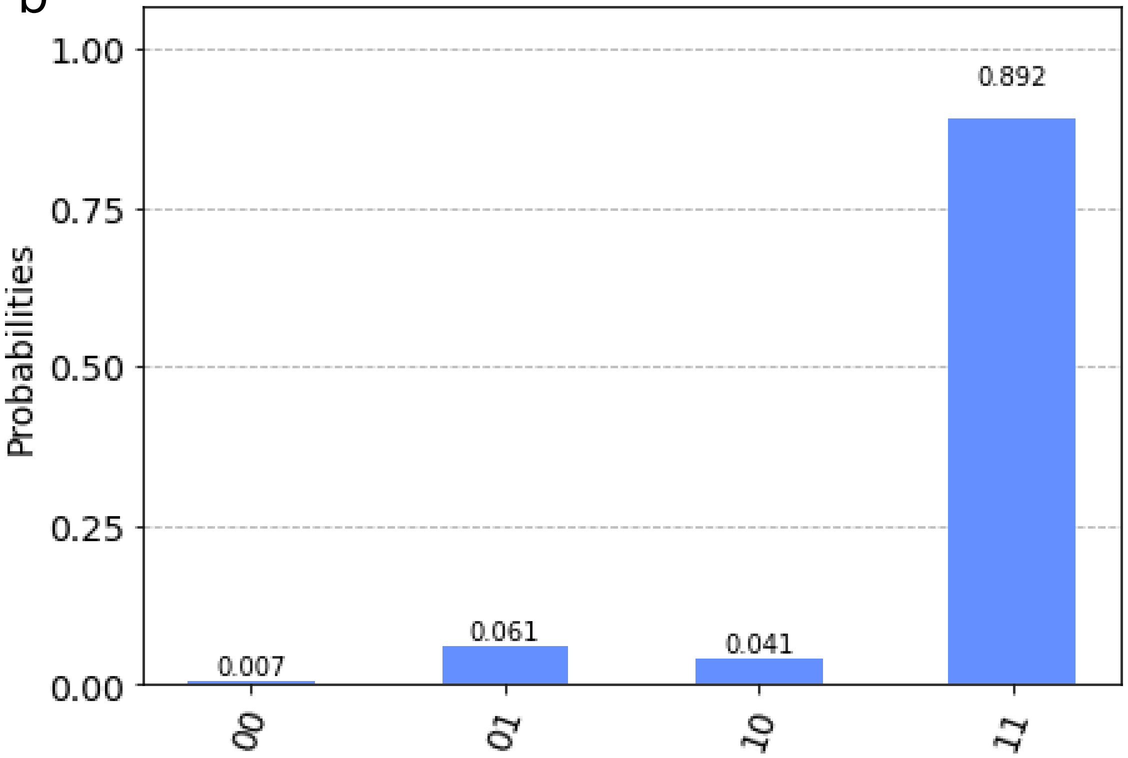
- Here are two executions of a Grover search. Can you guess which one was run on an actual quantum computer, and which one on an ideal simulator?

Actual

a



b



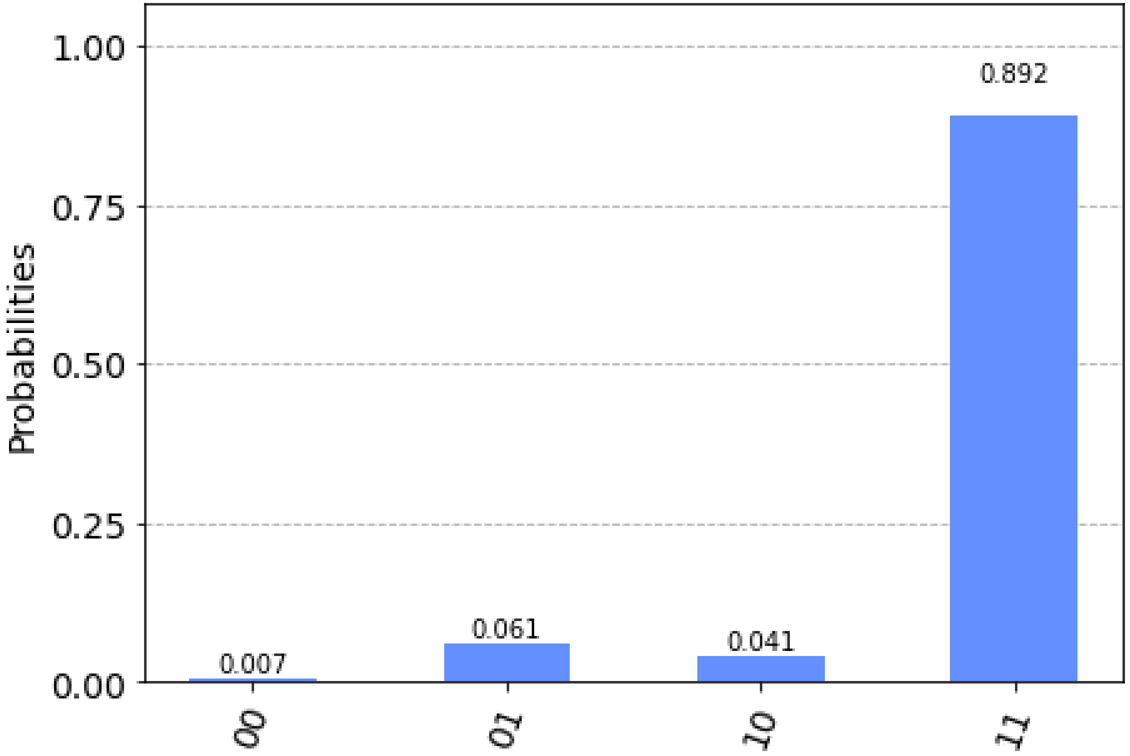
© 2021 The Coding School

DIFFERENT TYPES OF ERRORS

- What are all the different errors you can think of?

✓ • Relaxation
✓ • Decoherence
✓ • Phase shift
• Gate errors
• Ampl. errors
• Readout

- Qubit errors ← initialization
- Gate errors ← computation
- Readout errors ← output



© 2021 The Coding School

Canvas attendance quiz passcode: 8437

THE QC LANDSCAPE RIGHT NOW

- What is the status of QC right now?
  - We have quantum computers with tens of qubits * DWave
  - These QCs are noisy, make errors

- What can we do?
  - **Characterize errors (difficulty level - easy)**
  - Mitigate errors (difficulty level – medium)
  - Correct errors (difficulty level – hard)
  - Prevent errors (difficulty level – super hard)

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

CHARACTERIZING ERRORS - TELEPHONE

- Have you ever played the game of Telephone?
- You whisper a message to your friend, who whispers it to their friend, and so on
- Up to how many levels of friends can you reliably send a message?

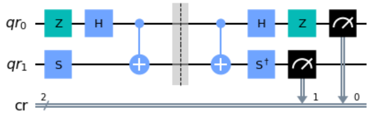
© 2021 The Coding School

Canvas attendance quiz passcode: 8437

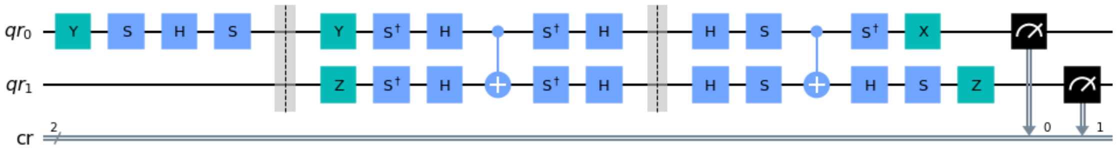
QUANTUM TELEPHONE

- Which circuit do you expect to have more errors?

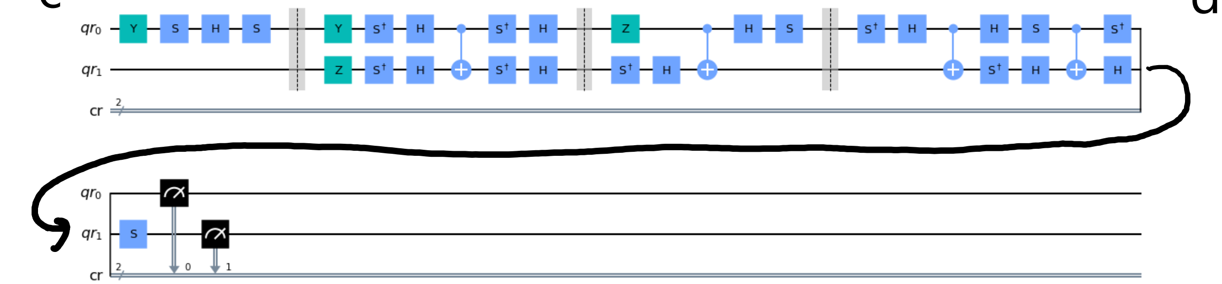
a



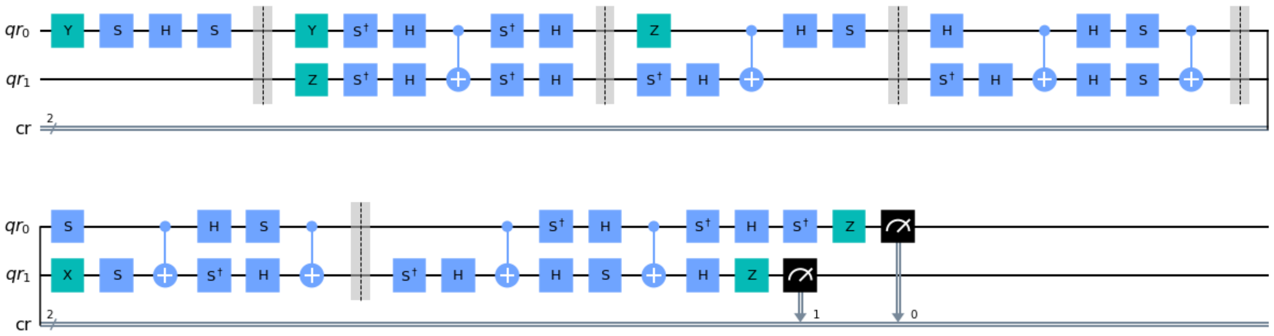
b



c



d



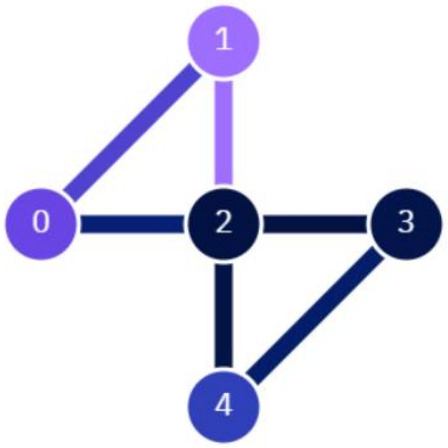
© 2021 The Coding School

Canvas attendance quiz passcode: 8437

USING DIFFERENT TELEPHONES

- On which quantum computer will there be fewer errors?

1) ibmq_5_yorktown



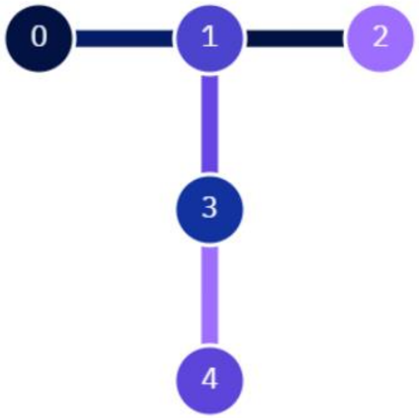
Avg. CNOT Error: 2.726e-2

Avg. Readout Error: 6.916e-2

Avg. T1: 53.74 us

Avg. T2: 33.09 us

2) ibmq_belem



Avg. CNOT Error: 2.448e-2

Avg. Readout Error: 3.674e-2

Avg. T1: 87.26 us

Avg. T2: 98.59 us

$|00\rangle \rightarrow \text{CNOT} \rightarrow |00\rangle$

$|01\rangle \rightarrow \text{CNOT} \rightarrow |01\rangle$

$|10\rangle \rightarrow \text{CNOT} \rightarrow |11\rangle$

$|11\rangle \rightarrow \text{CNOT} \rightarrow |10\rangle$

© 2021 The Coding School

Canvas attendance quiz passcode: 8437

RANDOMIZED BENCHMARKING

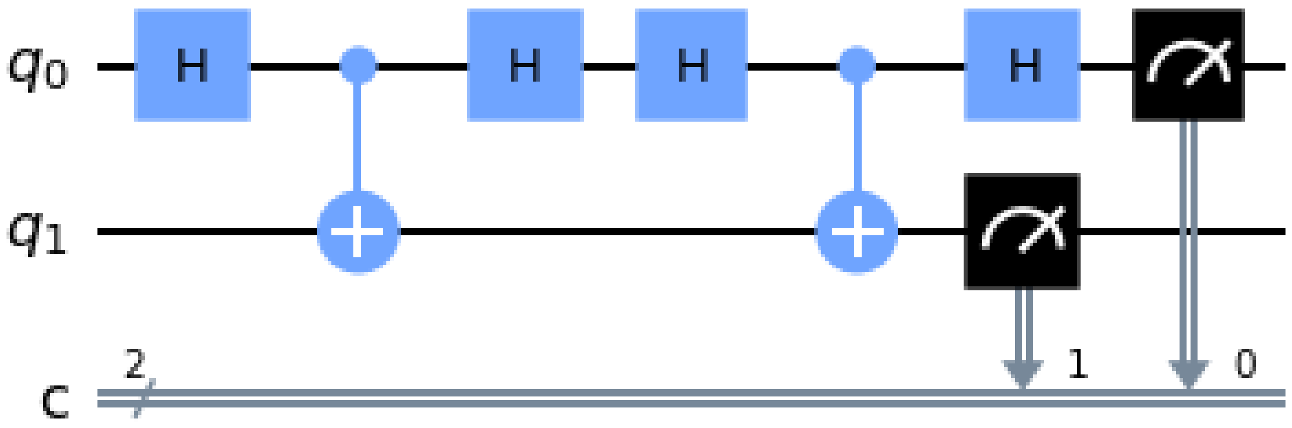
- Let's quantify our intuition – we'll simulate these circuits on ibmq_5_yorktown and ibmq_belem and see how many errors there are

- This process is called **randomized benchmarking**
  - The "message" we want to send is 00..0 – our circuits should ideally produce the state $|00..0\rangle$ as the result
  - We'll give the same circuit to both QCs to compare their performance
  - From the performance, we can extract the **average gate fidelity**

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

EXAMPLE OF BENCHMARKING CIRCUIT

- What is the output of this circuit?



© 2021 The Coding School
Canvas attendance quiz passcode: 8437

GENERATING RANDOM CIRCUITS

- Qiskit has a useful function that lets us generate random circuits of a given length

- This function is **randomized_benchmarking_seq()** in the library **qiskit.ignis.verification.randomized_benchmarking**

```
import qiskit.ignis.verification.randomized_benchmarking as rb
```

[1, 5, 10]

rb_circs, _ = rb.randomized_benchmarking_seq(*length_vector, rb_pattern*)

2-D list of circuits

another output that we ignore

list of circuit lengths

number and pattern of qubits

© 2021 The Coding School

Canvas attendance quiz passcode: 8437

INCLUDING NOISE MODELS IN QISKIT

- It will be really time consuming to run these circuits on actual QCs!
- Instead, we will use qasm_simulator, but add a **noise model** to it to mimic the noise of an actual QC

- Step 1: Importing NoiseModel

```
from qiskit.providers.aer.noise import NoiseModel
```

- Step 2: Choosing which QC's noise you want to mimic

```
backend = provider.get_backend('ibmq_belem')
```

- Step 3: Create a noise model from that QC

```
noise_model = NoiseModel.from_backend(backend)
```

- Step 4: Run a job using that noise model

```
job= execute(qc, backend = 'qasm_simulator', noise_model=noise_model, shots = 1024)
```


© 2021 The Coding School

Canvas attendance quiz passcode: 8437

STEPS IN RANDOMIZED BENCHMARKING

1. Create a noise model for the QCs we want to compare

```
backend_belem = provider.get_backend('ibmq_belem')
noise_model_belem = NoiseModel.from_backend(backend_belem)
```

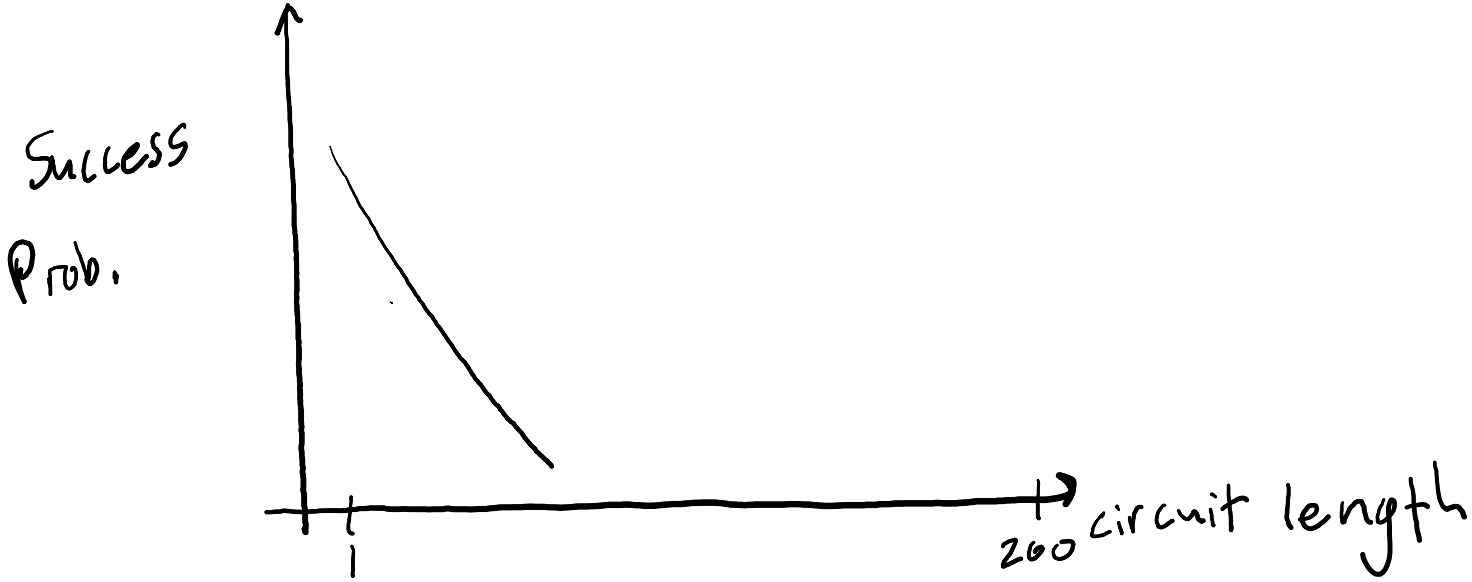

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

STEPS IN RANDOMIZED BENCHMARKING

2. Creating randomized circuits of different lengths

```
rb_circs, _ = rb.randomized_benchmarking_seq(length_vector = lengths, rb_pattern = pattern)
```

We will use lengths between 1 and 200, and we will simulate 3 qubit circuits



© 2021 The Coding School
Canvas attendance quiz passcode: 8437

STEPS IN RANDOMIZED BENCHMARKING

3. Running the circuits using the noise models from step 1

```
#getting results using the belem noise model
job_belem = execute(rb_circs[0][i], backend, noise_model=noise_model_belem, shots = sh)
results_belem = job_belem.result()
counts_belem = results_belem.get_counts()
count_000_belem[i] = counts_belem["000"] #extracting the 000 counts from belem results
```


© 2021 The Coding School
Canvas attendance quiz passcode: 8437

STEPS IN RANDOMIZED BENCHMARKING

4. Comparing results

```
plt.plot(lengths, count_000_belem/sh, marker = 's')
plt.plot(lengths, count_000_yorktown/sh, marker = 'o')
```


© 2021 The Coding School
Canvas attendance quiz passcode: 8437

TIME TO CODE!

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

KEY TAKEAWAYS

- Randomized benchmarking is the process of testing randomly generated circuits on different quantum processors to compare their error performance
- We can simulate the error performance of different processors in qiskit by using the NoiseModel function
- **Gate fidelity** can be extracted from the results of randomized benchmarking
- **Connectivity** is also an important metric in determining processor performance. If there is a direct connection between two qubits, we can apply two-qubit gates on them with fewer errors

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

FURTHER READING AND RESOURCES

- Qiskit textbook page on randomized benchmarking
- Qiskit textbook page on error correction
- Video on mitigating noise on real QCs
- Lecture on quantum error correction
- Introduction to decoherence
- Lecture on decoherence in quantum computers

© 2021 The Coding School
Canvas attendance quiz passcode: 8437

POST-LAB ZOOM FEEDBACK

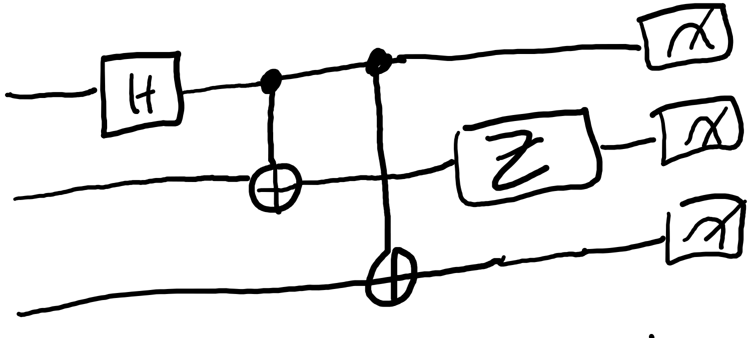
After this lab, on a scale of 1 to 5, how would you rate your understanding of this week's content?

- 1 –Did not understand anything
- 2 – Understood some parts
- 3 – Understood most of the content
- 4 – Understood all of the content
- 5 – The content was easy for me/I already knew all of the content

© 2021 The Coding School

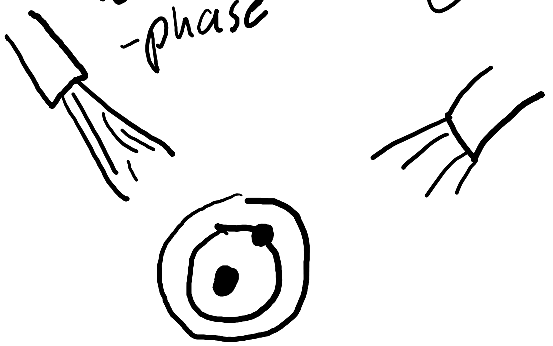
Canvas attendance quiz passcode: 8437

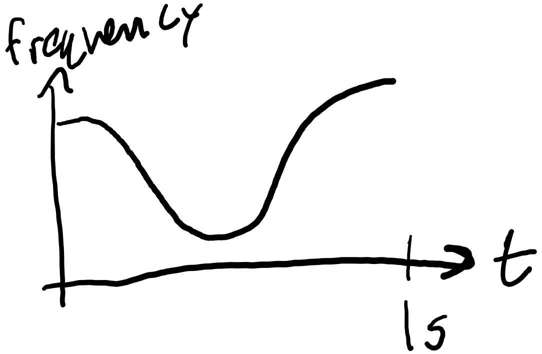
Qiskit
Python
Code

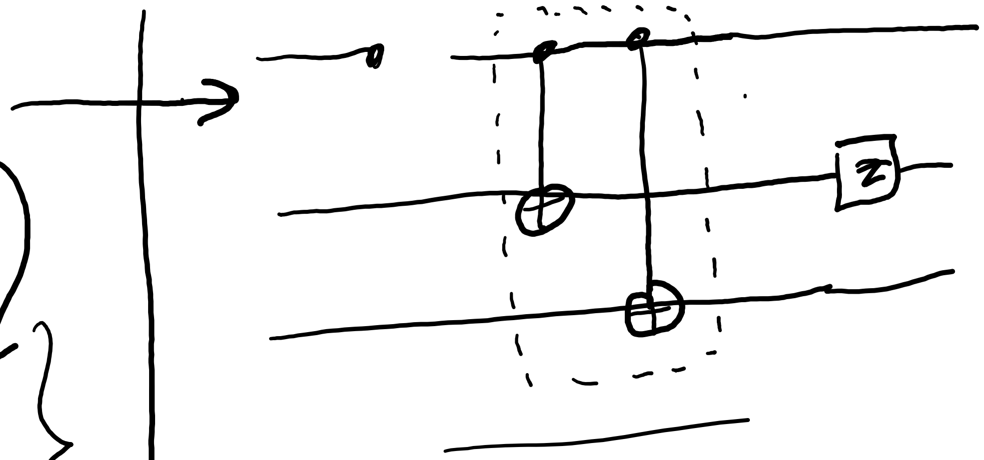


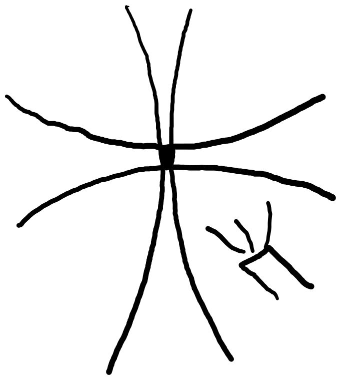
X	H	Z	CNOT
10MHz ✓	✓	✓	?
π/10MHz ✓	✓	✓	
O	✓	✓	

- frequency
- duration
- phase









© 2021 The Coding School
Canvas attendance quiz passcode: 8437