

# ECE 382V: Introduction to Quantum Computing Systems from a Software and Architecture Perspective

## Lab 1

Sayam Sethi

October 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Noisy Circuits</b>	<b>1</b>
2.1	Greenberger-Horne-Zeilinger Circuit . . . . .	2
2.2	Bernstein-Vazirani Circuit . . . . .	2
<b>3</b>	<b>Grover's Amplitude Amplification</b>	<b>6</b>
<b>4</b>	<b>Routing</b>	<b>6</b>
4.1	Routing and Swaps . . . . .	8
4.2	Noise Adaptive Routing . . . . .	8
4.2.1	FakeAthensV2 . . . . .	9
4.2.2	FakeAlmadenV2 . . . . .	9
4.2.3	FakeCambridgeV2 . . . . .	9
<b>5</b>	<b>Conclusion</b>	<b>10</b>

## 1 Introduction

Through the lab and the report, we explore the practical implications of the errors in the noisy circuits and how it affects various decisions that are made in the compilation process, especially in the NISQ era. We analyze various quantum algorithms such as the GHZ circuit, Bernstein-Vazirani circuit and also the Grover's algorithm which portray various quantum properties that are different from a classical machine. We will also make observations about the usability and correctness of the results obtained from running circuits on the IBM devices currently available to the public.

## 2 Noisy Circuits

For all the circuits, we use the Hellinger distance as the metric of success. The Hellinger distance is defined as follows:

$$H(\rho, \sigma) = \frac{1}{\sqrt{2}} \sqrt{\sum_i (\sqrt{\lambda_i} - \sqrt{\mu_i})^2} \quad (1)$$

Additionally, all circuits are run with 1024 shots. This is because the maximum number of qubits used is 5 and since we have  $(2^5)^2$ , it is almost guaranteed that the noise will be sufficiently captured. The statevector simulation is used as the ground truth for all the circuits.

## 2.1 Greenberger-Horne-Zeilinger Circuit

We implement the GHZ circuit by applying CNOTs between consecutive qubits. An example circuit using 5 qubits is shown in Figure 1. The GHZ circuit was run for different number of qubits on different devices with varying Quantum Volumes and the Hellinger distance with the statevector simulation is plotted in Figure 2.

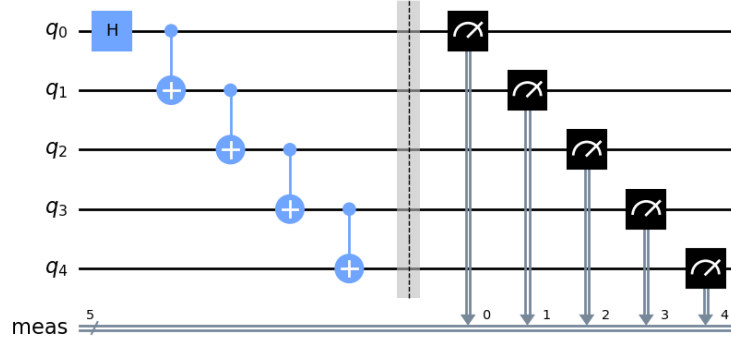


Figure 1: GHZ Circuit with 5 qubits

As clearly visible from the plots, the natural trend is that the Hellinger distance decreases with increasing Quantum Volume and increases with increasing the number of qubits. This is because the larger the Quantum Volume, the better is the quality of the result on the machine. Similarly, as the number of qubits and gates in the circuit increase, the total error of the circuit increases and hence the Hellinger distance is larger when the circuit has more qubits. However, **FakeQuitoV2** is an outlier for all the plots. This can easily be inferred from the error map of the machine (Figure 3) which has a much higher error rate than **FakeLimaV2** and the other machines.

## 2.2 Bernstein-Vazirani Circuit

An example circuit used for modelling the performance of the Bernstein-Vazirani circuit is shown in Figure 4. The plots for the Bernstein-Vazirani circuit are shown in Figure 5.

The trend observed is quite similar to the trends observed in the GHZ circuit. The reason for **FakeQuitoV2** having a higher error rate than is the same as explained in Section 2.1. Otherwise, the Hellinger distance decreases as the Quantum Volume increases which is what we expect. However, the trend for increasing the number of qubits is not so apparent. This is because the circuit complexity does not increase with the increase in the number of qubits. The circuit with 4 qubits has a much simpler bit-string as the secret due to which the total error does not increase as much in comparison to the circuit for 3 qubits. In fact, the Hellinger distance for the 4 qubit circuit is lower than the 3 qubit circuit for **FakeKolkataV2**. This is because 5 qubits is 2 lower than  $\log 128$ , which is the maximum number of qubits that were used to get the heavy strings when determining the Quantum Volume of **FakeKolkataV2**. Therefore, the error is lower when lesser number of gates are used on **FakeKolkataV2**. This is not the case for the other circuits since the number of qubits used is close to the number of qubits used to compute the Quantum Volume and hence the errors grow considerably.

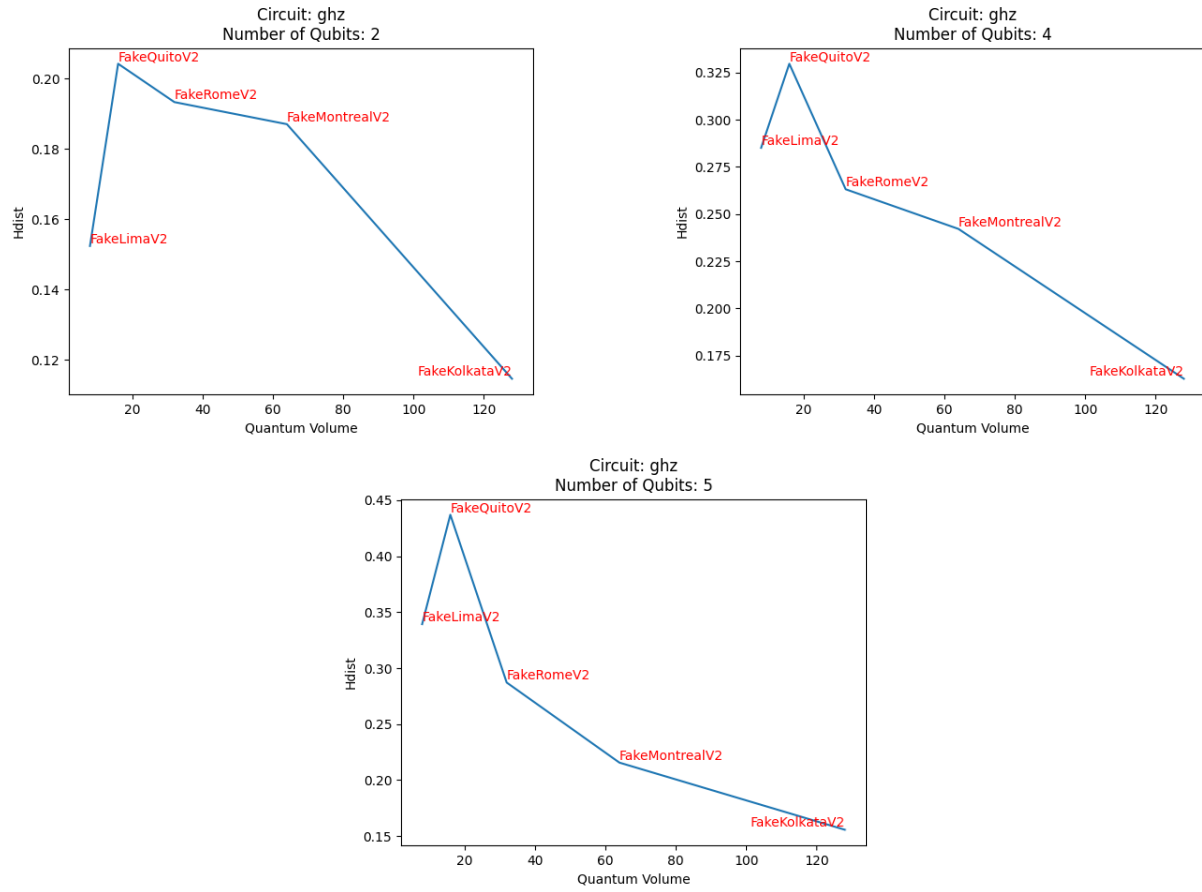


Figure 2: Plots of Hellinger distance between statevector simulation and different Quantum Volumes

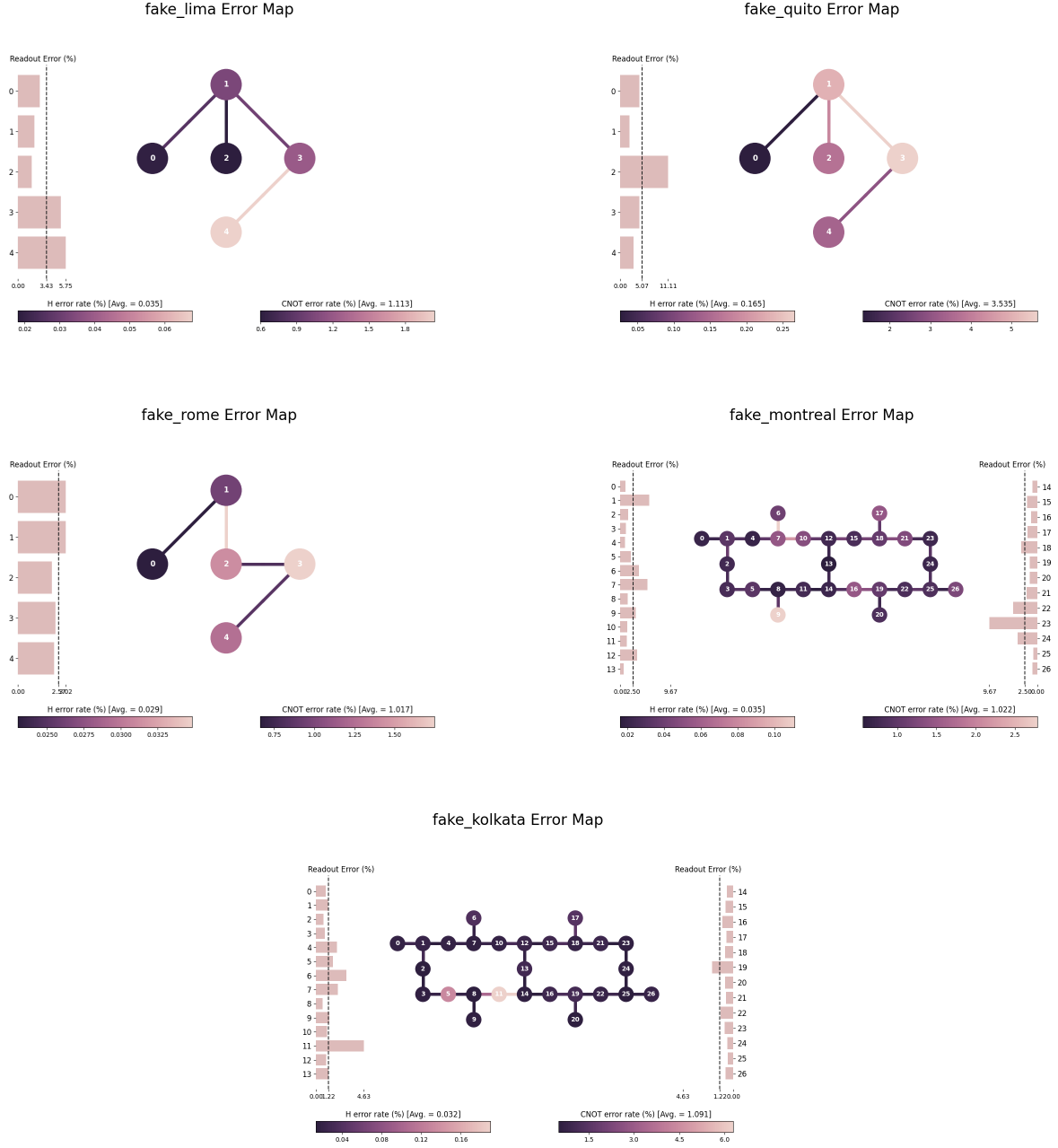


Figure 3: Error maps of the different backends

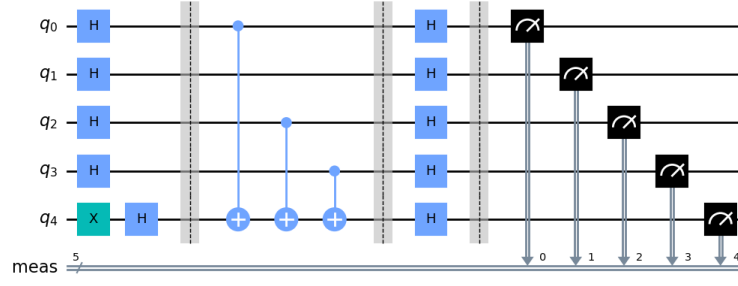
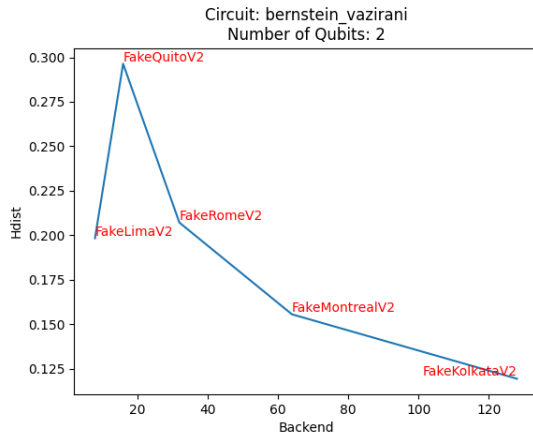
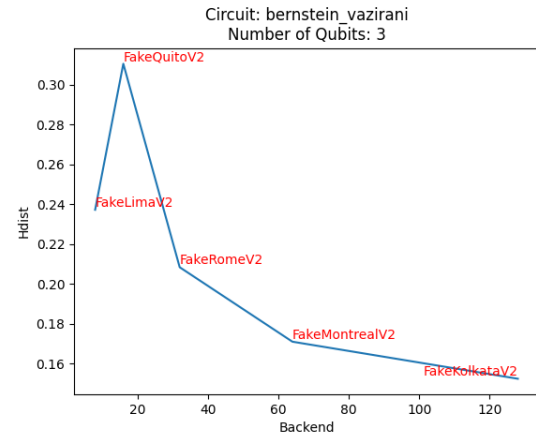


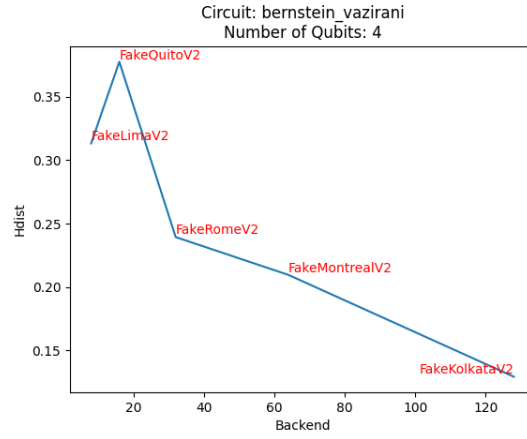
Figure 4: Bernstein-Vazirani Circuit with 4 qubits and an ancilla qubit (secret used in 1011)



(a) Secret = 11



(b) Secret = 110



(c) Secret = 0001

Figure 5: Plots of Hellinger distance between statevector simulation and different Quantum Volumes

### 3 Grover's Amplitude Amplification

For this circuit, the metric used is the Hellinger distance between the distribution with a 1 for the solution bit-string in the database and the distribution obtained from running the circuit on the machine (either statevector or `FakeKolkataV2`). Again, the circuit was executed for 1024 shots. The optimal number of iterations for 5 qubits would be,

$$\left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} - \frac{1}{2} \right\rceil = \left\lceil \frac{\pi}{4} \sqrt{\frac{2^5}{1}} - \frac{1}{2} \right\rceil = 4 \quad (2)$$

The Oracle is defined as  $\mathbf{I}$  (where  $\mathbf{I}$  is the identity matrix) but a  $-1$  in the position corresponding to the solution bit-string. The diffuser is defined as  $-\mathbf{I}$  but a 1 for the first row and column.

A sample circuit for solving the Grover's problem is shown in Figure 6. The plots for the Grover's circuit are shown in Figure 7.

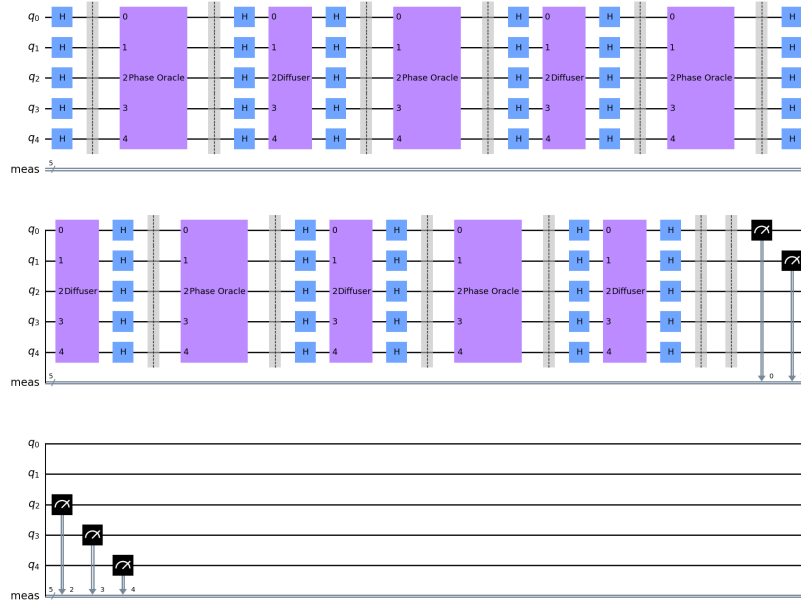


Figure 6: Grover's Circuit with 5 qubits and 5 iterations

The results for the statevector simulation are exactly what they would be theoretically, and we get a Hellinger distance of exactly 0 for 4 iterations which verifies our computation. However, for the noisy case, we never get a good result and the output is very noisy for all the iterations. This is because the oracle and diffuser is a 5-qubit gate which is decomposed into multiple 2-qubit gates and this greatly increases the error rate of the circuit. The lowest Hellinger distance that we get is for 42 iterations which is still very large and close to 0.89 (which is essentially just noise).

### 4 Routing

The error maps for the three architectures considered are shown in Figure 8. The error map also gives us an idea of the connectivity of the different architectures.

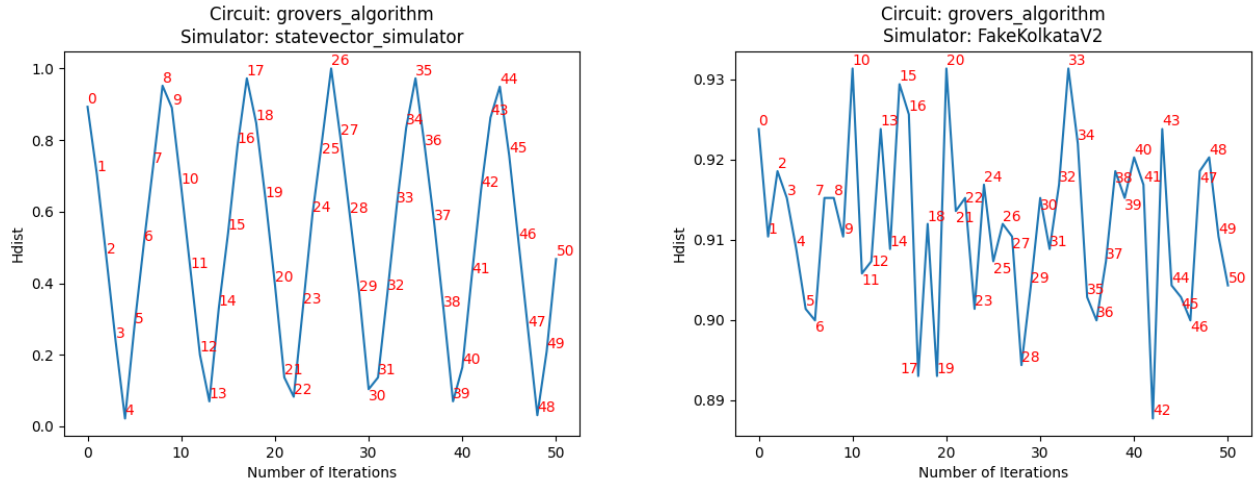


Figure 7: Plots of Hellinger distance for statevector and FakeKolkataV2

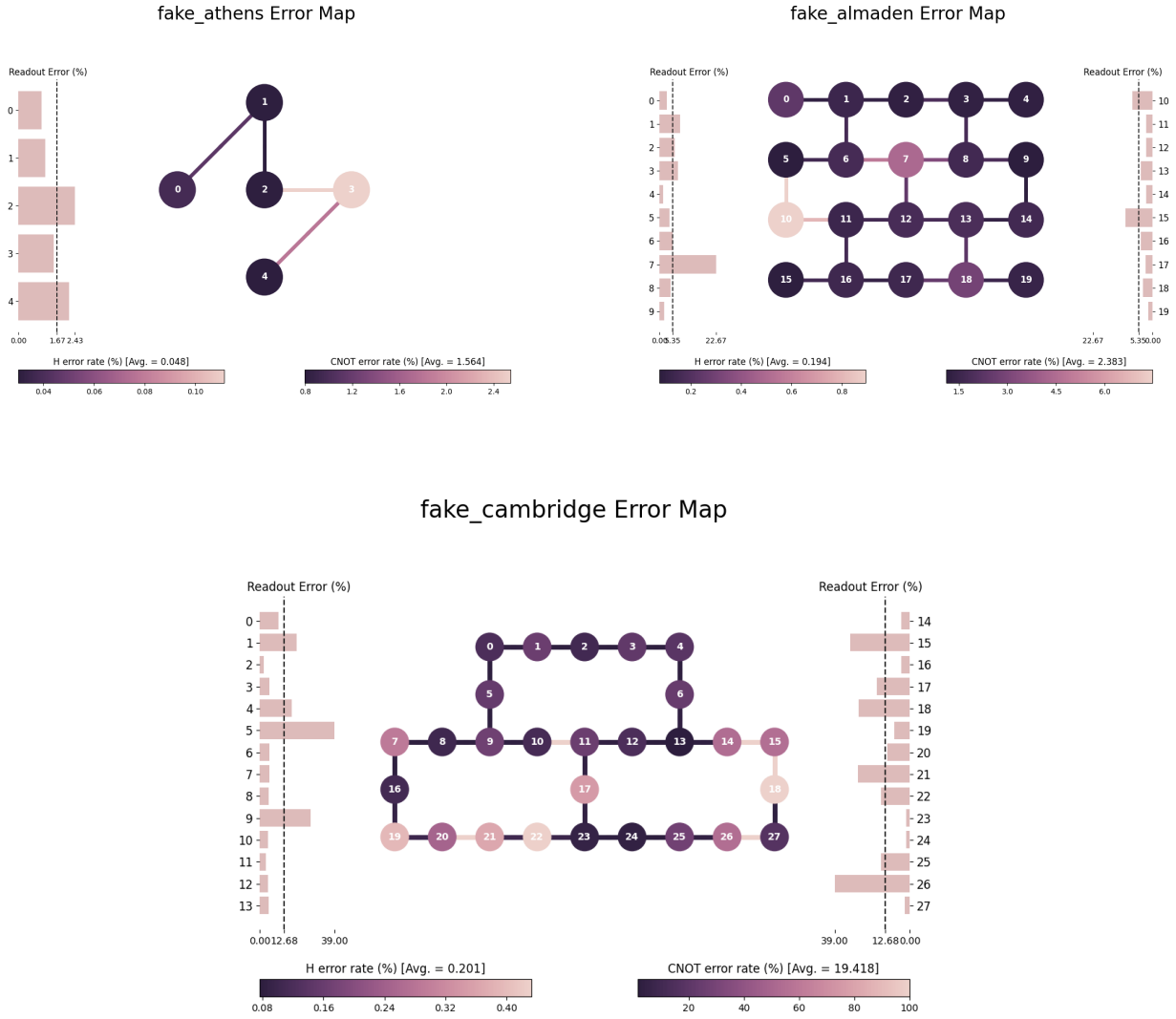


Figure 8: Error maps for different architectures

## 4.1 Routing and Swaps

The plots for the number of swaps required to execute the `ComplexCircuit` on the different architectures are shown in Figure 9.

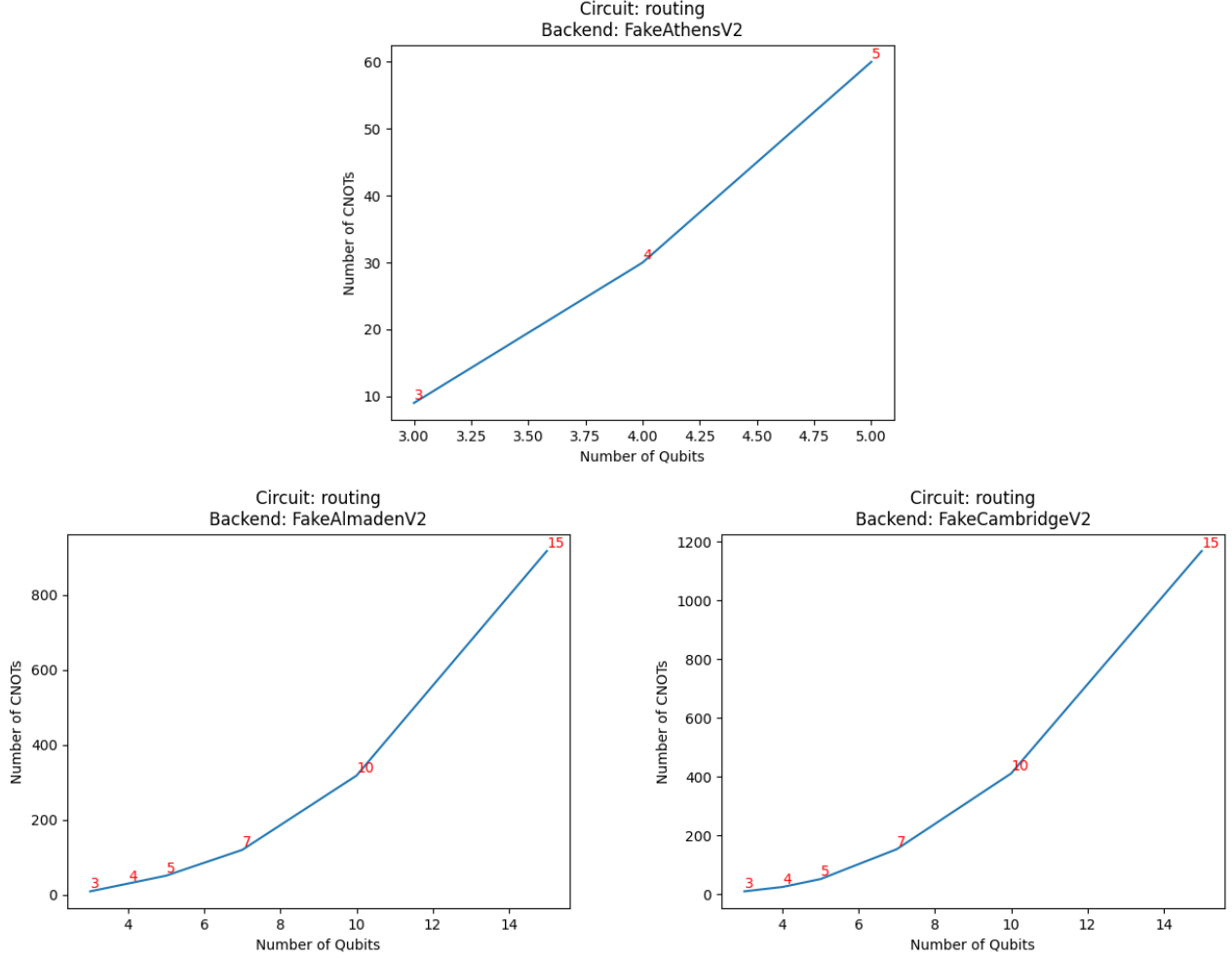


Figure 9: Number of swaps required for `ComplexCircuit` on different architectures

The number of swaps required for `FakeAlmadenV2` is the least since its architecture has the maximum connectivity. On the contrary, `FakeCambridgeV2` has the highest number of swaps required since there is limited connectivity among the qubits even though it has more qubits. The number of swaps required for `FakeAthensV2` is almost the same as `FakeCambridgeV2` since the connectivity for the first 5 qubits is the same for both the architectures. The number of swaps required for `FakeCambridgeV2` is slightly lesser since it has more qubits to allow for slightly more flexibility, and it doesn't need to always work with just 5 qubits in total.

## 4.2 Noise Adaptive Routing

For all architectures, the Bernstein-Vazirani circuit was chosen with the secret key 1111 to introduce the maximum number of CNOTs and hence be able to easily differentiate between the algorithms used at different optimization levels.



#### 4.2.1 FakeAthensV2

The mappings for different optimization options are shown Figure 10. Data qubit 4 is the only qubit that has CNOTs with all other qubits and hence it should be placed in a position with low CNOT error rates. Therefore, in `optimization_level 1` data qubit 4 is placed in physical qubit 1 of the architecture. However, `optimization_level 2` and `3` instead place data qubit 4 in the position of logical qubit 2 since this leads to reduced number of SWAP insertions even though it increases the actual CNOT error rate for a few gates in the original circuit, thus reducing the combined error rate of the circuit.

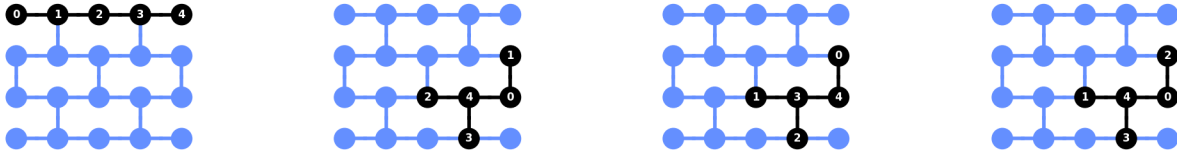


(a) `optimization_level 0` (b) `optimization_level 1` (c) `optimization_level 2` (d) `optimization_level 3`

Figure 10: Mappings for different optimization levels for FakeAthensV2

#### 4.2.2 FakeAlmadenV2

The mappings for different optimization options are shown Figure 11. All the (non-zero) optimization levels try to place the data qubits so that there is maximum connectivity between data qubit 4 and the other qubits while also ensuring that the CNOT error rates are low. Similar to the previous part, `optimization_level 1` optimizes for reduced CNOT error rates. `optimization_level 2` optimizes for reduced number of swaps and `optimization_level 3` balances between the two.



(a) `optimization_level 0` (b) `optimization_level 1` (c) `optimization_level 2` (d) `optimization_level 3`

Figure 11: Mappings for different optimization levels for FakeAlmadenV2

#### 4.2.3 FakeCambridgeV2

The mappings for different optimization options are shown Figure 12. The placement is chosen so that the two-qubit gates have the least errors and the minimal number of swaps are introduced. `optimization_level 1` chooses the arrangement that minimizes the individual CNOT error rates. `optimization_level 2` chooses the placement that minimizes the number of swaps and hence leads to a hole between the qubits since whichever qubit has to perform a CNOT with the data qubit 4 can swap with the empty qubit which would

require one lesser CNOT gate. Finally, `optimization_level 3` balances between the two and hence places data qubit 4 in the most accessible position and in the middle of the other qubits.



(a) `optimization_level 0` (b) `optimization_level 1` (c) `optimization_level 2` (d) `optimization_level 3`

Figure 12: Mappings for different optimization levels for FakeCambridgeV2

## 5 Conclusion

Thus, in the lab we analyzed various Quantum Algorithms and the routing and placement problem for quantum compilers. We observed that the quantum computers at their current state cannot be used even for some of the simple circuits that can be easily computed classically (since the sizes are very small). However, we also observed that the different routing and placement algorithms that take advantage of the device level metrics strive towards reducing the errors and are quite successful in doing so, at least for small devices since the search spaces are small. This also leads us to realize that there is plenty of scope for research in this domain so that there are state-of-the-art compilers by the time we have devices that have enough qubits to actually run circuits with practical advantages.