A Software Method For Mitigating Single Qubit Errors On Superconducting Quantum Devices

Macauley Coggins[1]
mcoggins@quantumcomputinguk.org
Quantum Computing UK

Devanshi Arora[2]
darora@quantumcomputinguk.org
Quantum Computing UK

***Abstract*— Quantum error correction schemes have gained a lot of attention in recent years. This is due to the emergence of small scale quantum devices that make use of superconducting qubits. However these devices are noisy and prone to quantum decoherence and thus errors. Along with quantum error correction there has been a push for new schemes in quantum error mitigation that take a more passive approach in eliminating readout errors. In this research we introduce a software method for quantum error mitigation that maps virtual qubits in a circuit to physical qubits with the least error. The method developed was tested on 9 IBM quantum devices. Results in the study have shown the method can reduce readout errors by up to 35.52%.**

I. Introduction

Over the last several years quantum hardware has matured to the point that many companies and institutions have made their quantum devices open to public access. The majority of these devices use a type of superconducting qubit known as the charge qubit. This is a type of qubit that utilises a cooper pair box (CPB) which consists of a small superconducting island connected to a reservoir via a tunnel junction such as a Josephon junction. One of the major challenges involving implementation of charge qubits is short dephasing times (T_2) and relaxation times (T_1). This is mainly due to charge noise in electrostatic potential. Several approachable methods for mitigating this problem have been studied. One study by Vion et al found that operation of charge qubits at an optimal operating point called the 'sweet spot' corresponding to a specific charge degeneracy point through biasing of the qubit can increase dephasing times[1]. In 2007 a major breakthrough came with the development of the transmission-line shunted plasma oscillation qubit or Transmon qubit for short by Koch et al[2]. This charge insensitive qubit was designed to operate in increased ratio of Josephson energy and charging energy,E_J/E_C. This reduces sensistivity to charge noise along with an increase in qubit-photon coupling. A major feature of the Transmon is that the two superconductors are shunted capacitively. This results in a reduction of charge noise and increased coherence times. The hamiltonian can be stated as:

$$\hat{H} = 4E_c(\hat{n} - n_g)^2 - E_j cos\hat{\varphi} \tag{1}$$

where n_g is the effective offset charge of the device, E_c is the charging energy, E_j is the Josephson energy and $\hat{n}$ denotes the number of Cooper pairs transferred between the islands and $\hat{\varphi}$ denotes the gauge-invariant phase difference between the superconductors.

A. Quantum Error Correction

However despite recent breakthroughs decoherence and dephasing is still an issue that leads to errors. As such there is a push to develop quantum error correction schemes.

One of the simplest error correction schemes is to simply run the quantum circuit a number of times to get a probability distribution such that the result with the highest probability is the correct result. However this is very redundant and time consuming especially when current quantum devices rely on time sharing. A more active but simple method of quantum error correction is the bit flip code [3]. This is a simple scheme that makes use of 2 ancillary qubits to correct bit flip errors on 1 qubit. For N qubits the method applies CNOT gates to q1 and q2 where the control qubit is q0. Then this is repeated and then a toffoli gate is applied to q0 where q1 and q2 are the control qubits.

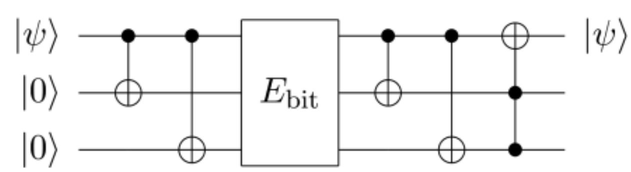

Fig. 1. Diagram of a 3 qubit bit flip error correction circuit

Another form of quantum error is the phase error. This is where the phase of the qubit has become corrupted. If a qubit is operated purely in the computational basis then this would not be a problem however the majority of quantum algorithms make use of the hadamard basis and thus phasing for general operation. To solve a phase flip error the 3 qubit bit flip code can be used with the addition of hadamard gates[4].

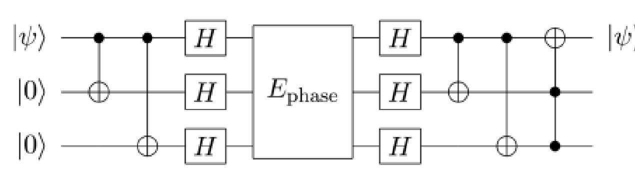

Fig. 2. Diagram of a 3 qubit phase flip error correction circuit

While quantum error correction schemes can be used to solve both bit and phase errors they require ancillary qubits. Indeed for the circuits above 2 ancillary qubits are required for every input qubit. If the Shor code is used for solving bit/phase flip errors then 8 ancillary qubits are required for every 1 input qubit[5].

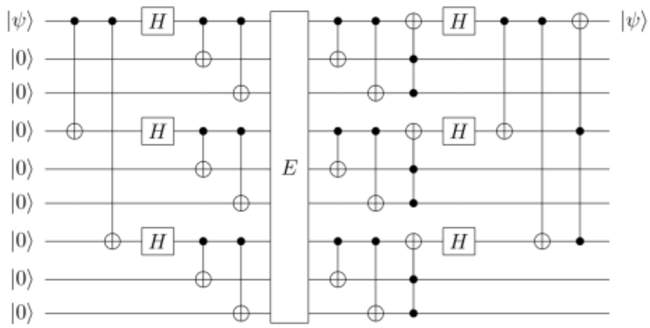

Fig. 3. Diagram of the Shor code circuit

B. *Quantum Error Mitigation*

As current quantum devices have a very low qubit count the use of ancillary qubits for use in error correction should be used sparingly. As such another avenue for solving errors is quantum error mitigation. This is a more passive approach where the possibility of errors is reduced wherever possible without actively correcting any errors that may arise. Various hardware and software methods have been developed to mitigate quantum errors. Endo et al developed a mitigation method that eliminated Markovian errors with single qubit clifford gates[6]. In another study by Song et al a mitigation protocol was developed that utilised gate set tomography and quasi-probability distribution [7]. A scheme by Zhang et al for circuit optimization involves using a software algorithm to exchange qubits based upon the layout of the device by using SWAP gates. Furthermore the scheme also further optimized the circuits by merging single qubit gates when necessary and as such reduces the probability of errors [8]. A hardware method by Otten et al took a different approach. In their study results were combined from many slightly different experiments. This method increased times for spontaneous emission T_1 and dephasing T_2. Furthermore the method overhead was based solely on the number of evaluations [9].

II. Preliminary Materials

For this research a range of software and hardware were used.

A. *Qiskit*

Qiskit is an open source software development framework for working with quantum computers including quantum devices from IBM. Qiskit contains a module called Terra which contains a set of functions for creating quantum circuits and transpiling. A Quantum circuit is a model for creating quantum algorithms where each circuit is represented as a sequence of gate operations on N qubits[10]. In this research the qiskit transpile function was used to map the logical qubits in a ciruit to the physical qubits on a quantum device with the least read out error [11].

B. *IBM Quantum Devices*

For testing and verification of the software method 9 quantum devices from IBM were used. These devices make use of superconducting transmon qubits. Each device used has a relatively low qubit volume ranging from 5 to 15 qubits. The list of quantum devices used can be found in Table 1. One device called IBMQ Armonk was not used as only 1 qubit is present on the device. Calibration data on the 9 IBM quantum devices used can be found in the supplementary data section.

Name	Qubit volume
IBMQ Athens	5
IBMQ London	5
IBMQ Burlington	5
IBMQ Essex	5
IBMQ Ourense	5
IBMQ Vigo	5
IBMQX2	5
IBMQ Rome	5
IBMQ 16 Melbourne	15

TABLE I

III. Software Method

The software method aims to mitigate errors by assigning virtual qubits in a circuit to the physical qubits on a device with the least error. First a circuit was created as seen in Fig 4. This simple circuit consists of 3 qubits and applies a Pauli-X gate to all of qubits and then each qubit is measured. The X gate is a very useful gate that flips the state of the qubit in the computational basis from $|0\rangle$ to $|1\rangle$ and vice versa. As such when the circuit is ran on a quantum device it should be seen that all qubits that were first initialized to $|0\rangle$ should be $|1\rangle$.The expected register measurement should be $|111\rangle$ after the X gate is applied. Thus it is a simple way to measure qubit fidelity in terms of readout errors and gate errors.

After this circuit is initialised it is passed to the mitigation function as described in Algorithm 1 along with the Threshold value, mode value, and backend object. In the algorithm 3 arrays are initialised. Q_e which holds the physical qubit error values. Q_q which holds the physical qubits and Q_m which is the virtual to physical qubit map. Next for each qubit the error is measured. If the mode passed to the function is 0 then the read out error of the qubit is used. If the mode is 1 then the U3 gate error is used instead. If the specified error is less than or equal to the the threshold value then it is appended to Q_e. The qubit object is then appended to Q_q.

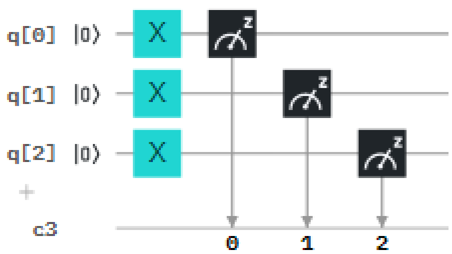

Fig. 4. Diagram of circuit used. q[0-2] are the qubits and c3 is the classical channel for read-out results.)

Next for each error value in Q_q it is sorted along with the qubit in Q_q from smallest to largest.

After the selected qubits have been sorted the virtual to physical qubit map Q_m is created by appending the first N qubits to the map where N is the number of qubits in the circuit.

Result: Returns a Logical to Physical qubit map Q_m for use in transpiling
Data: Threshold,Mode,Backend,Circuit
for *Qubit q in Backend* **do**
  $R_r \leftarrow$ `Readout Error`
  $R_g \leftarrow$ `U3 Gate Error`

  if *Mode = 0 and* $R_r <= Threshold$ **then**
    $Q_e.append(R_r)$
  else
    if *Mode = 1 and* $R_g <= Threshold$ **then**
      $Q_e.append(R_g)$
    end
  end
  $0_q.append(Qubit)$
  for *i in* Q_e **do**
    Sort O_q by Q_e from smallest to largest
  end
end
for *i in* $Q_{circuit}$ **do**
  $Q_m.append(O_q[i])$
end
Return Q_m

Algorithm 1: Mitigation function

IV. Results

In order to record the performance of the mitigation method 3 variations of the circuit described in fig 4 were ran on 9 IBM quantum devices 8192 times each. The first circuit was the unoptimised circuit as seen in fig 4. The second circuit was the same but used the mitigation function to use qubits with the lowest read out error. The the third circuit used the mitigation function to use qubits with the lowest U3 gate error.

Overall the results from the experiments are very promising. In the best case scenario on the IBMQ London device the software method devised outperformed the normal circuit by 35.52% when readout optimised and 35.22% when optimised against gate error (U3) values. In the worst case scenario on the IBMQX2 device the unoptimised method outperformed the software method by 1.01. However when the software method was optimised against gate error values it outperformed the unoptimised method by 0.145%. Further comparison of the readout and U3 gate errors across all qubits on the IBMQX2 device shows low variance in error values between devices with 1.283e−4 across readout errors and 2.405e−8 across U3 gate errors as seen in in Table 2. Furthermore on the IBMQ London device there was higher variation in error values with 9.383e−3 for readout errors and 3.707e−3 for U3 gate error values as seen in Table 3. From the results in variance it can be seen that the software method developed in this research is not effective if variance in errors across qubits is low. Furthermore in the best case scenario the software method outperforms unoptimised circuits considerably when there is high variance across qubit errors on a device and in the worst case scenario it will perform around as well as the unoptimised circuit if the variance is low. A histogram comparing the performance of the unoptimised and optimised circuits across all devices can be found in Fig 5. Calibration data on the IBM devices used can be found in the supplementary data section.

Qubit	Readout error	U3 gate error
Q0	0.007	0.00124
Q1	0.0155	0.0014
Q2	0.028	0.001
Q3	0.0275	0.00135
Q4	0.0355	0.00121
Variation	0.000128325	0.00000002405

TABLE II
IBMQX2 Readout and U3 gate errors

Qubit	Readout error	U3 gate error
Q0	0.01667	0.0014
Q1	0.245	0.13748
Q2	0.045	0.00192
Q3	0.03	0.00118
Q4	0.02667	0.00083
Variation	0.00938399667	0.00370738472

TABLE III
IBMQ London Readout and U3 gate errors

V. Discussion

From the results the mitigation method developed in this paper shows promise with an increase in performance of up to 35.52% on certain quantum devices. However it was also seen that the method provides no performance increase if variance between qubit errors is low. As such this method is recommended to be used on devices where variance between qubit errors is high. Given that this is a passive error mitigation method it has a large advantage in that it does not require ancillary qubits as is needed in error

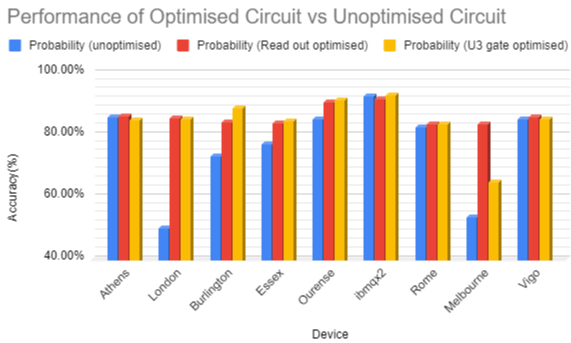

Fig. 5. Histogram detailing accuracy of unoptimised circuit vs optimised circuit on different devices

correction methods. Considering the low qubit volume of present superconductor based quantum devices this is a huge advantage over quantum error correction methods as qubit counts are limited. Given that this this method has been developed for mitigating readout errors and single gate errors future research will look to develop this method further to mitigate multi qubit gate errors.

VI. Conclusion

In conclusion this paper has shown a novel error mitigation method for superconducting quantum devices. Results from this paper show that circuits optimised with this method have a higher accuracy compared to unoptimised circuits. Results also show that the method performs best on devices where there is a high variance across qubit errors. Future research will aim to develop this method further to mitigate errors on multi qubit gates.

VII. Supplementary Data

Source code for the quantum mitigation method and calibration data of the IBM quantum devices used can be found on OSF: https://osf.io/xqsd2/

References

[1] D. Vion. Manipulating the quantum state of an electrical circuit. *Science*, 296(5569):886–889, May 2002.

[2] Jens.et al. Koch. Charge insensitive qubit design derived from the cooper pair box. *Phys. Rev. A 76, 042319*, 2:1–11, 09 2007. arXiv:cond-mat/0703002v2.

[3] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, Jun 2013.

[4] L. Tornberg, M. Wallquist, G. Johansson, V. S. Shumeiko, and G. Wendin. Implementation of the three-qubit phase-flip error correction code with superconducting qubits. *Physical Review B*, 77(21), Jun 2008.

[5] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[6] Suguru Endo, Simon C. Benjamin, and Ying Li. Practical quantum error mitigation for near-future applications. *Physical Review X*, 8(3), Jul 2018.

[7] Chao Song, Jing Cui, H. Wang, J. Hao, H. Feng, and Ying Li. Quantum computation with universal error mitigation on a superconducting quantum processor. *Science Advances*, 5(9), 2019.

[8] Xin Zhang, Hong Xiang, Tao Xiang, Li Fu, and Jun Sang. An efficient quantum circuits optimizing scheme compared with qiskit, 2018.

[9] Matthew Otten and Stephen K. Gray. Recovering noise-free quantum observables. *Phys. Rev. A*, 99:012338, Jan 2019.

[10] Quantum circuits (qiskit.circuit) - qiskit 0.19.1 documentation, May 2020. https://qiskit.org/documentation/apidoc/circuit.html.

[11] Transpiler (qiskit.transpiler) - qiskit 0.19.1 documentation. https://qiskit.org/documentation/apidoc/transpiler.html.