Project Update: Quantum Computing Fall 2022
Circuit Debugging with Breakpoints

Palvit Garg, Harshwardhan Joshi , Shawn Salekin

Progress Summary

In the project proposal, we mentioned attempting two different approaches. However, we realized that approach 1 was trivial but inefficient, so we focused our attention to approach 2. This approach involved measuring the circuit at breakpoint, saving that as a unitary matrix, and resynthesizing the circuit from that unitary to resume execution. Up until now, we conducted literature review and experiments to assess the feasibility of this approach.

Our findings indicate that including phase information is necessary and can be generated using simulation runs. We ran experiments and synthesized an equivalent circuit from a unitary matrix and resumed the circuit from that point. Experiment-wise, our focus is to figure out how to generate (and under what condition) a unitary from hardware - all our attempts with experimental circuits have been unsuccessful.

Link to Project's GitHub Page

Milestone

Following is a detailed plan for the upcoming weeks and open questions that need to be addressed.

TaskId	Task	Estimated Completion Date	Steps Necessary to Achieve Task /Comments
1	Investigate whether it is possible to generate a unitary matrix from a hardware run	Next Week	Discussion with professor, More experiments ran on hardware
2	If not, then the only possible solution exists by using simulation and actual hardware parallelly. If yes, then we can use the unitary matrix from hardware itself.	Next Week	(dependent on task 1)
3	Validate whether the phase data obtained from simulation(s) can be used in synthesizing an equivalent circuit after a breakpoint	2 Weeks from now	Literature review, validation with experimental data from hardware runs
4	Figure out a method to extract phase data from parallel simulation runs	2 weeks from now	(dependent on the outcome of task 3)

5	Implement a qiskit module/extension that would allow qiskit users to add breakpoint(s) to their circuit.	2 Weeks from now	
6	Add tests and run a number of experiments to validate the correctness of our implementation	3 Weeks from Now	Could use famous QC algorithms/circuits for this task.

Solved Task(s)

During this time period, we spent a significant amount of time in research as well as running experiments to validate our ideas and/or findings.

1. Measuring a circuit mid-execution loses phase information. So we had to figure out how crucial the phase information was. Our finding is that for some circuits, we could lose the phase information; however, some other circuits/algorithms rely heavily on relative phases of the circuits - so we ultimately need the phase information. [**Shawn**]
2. To alleviate the issue with lost phase data, we proposed running a parallel circuit on a simulator, as the phases of these circuits can potentially be calculated deterministically. [**Shawn]**
3. Figured out different ways to generate unitary matrix from simulation run and synthesize a circuit from those - however, generating a unitary matrix from hardware runs have been unsuccessful so far. [**Harshwardhan]**
4. We are able to synthesize a new circuit using a unitary matrix (generated using a simulator) from a previous run on hardware. [**Palvit**]

Open Task(s)

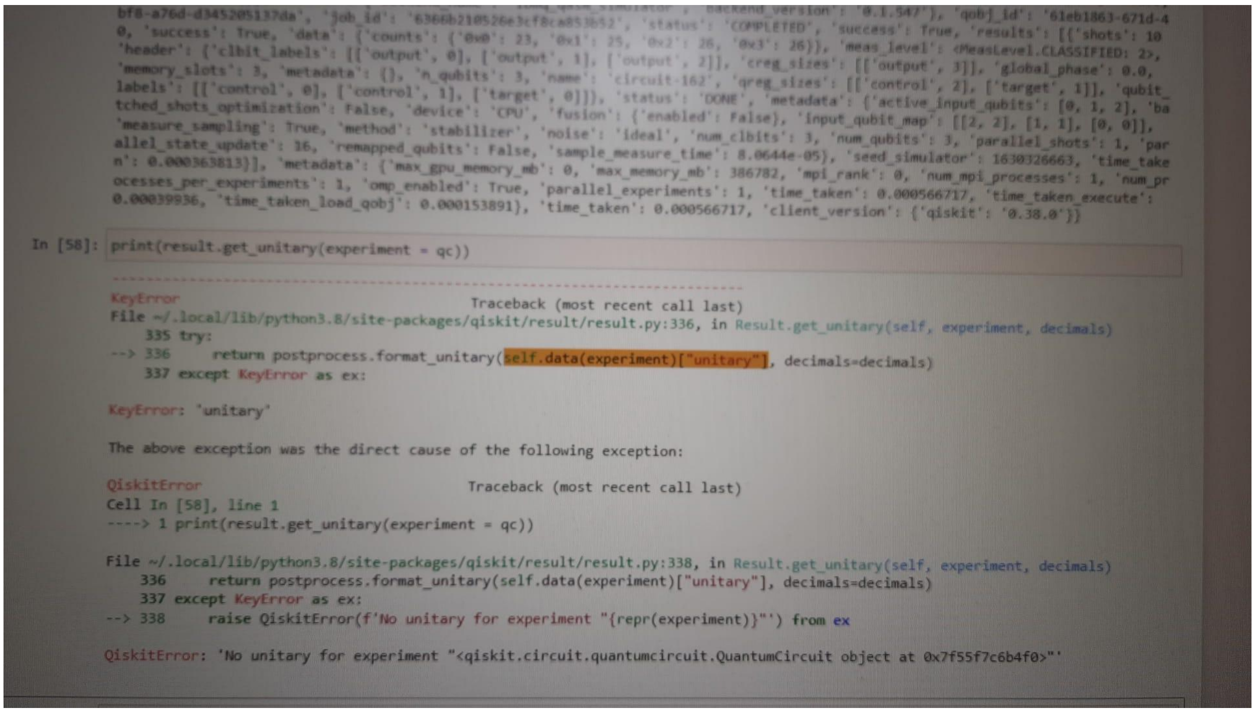
1. We need to figure out if there is any way to find the unitary matrix from hardware run. All three of us had tried generating unitary matrices on real hardware but failed.
2. Why do the recreated circuits don't reproduce the same results as simulation?
3. If we are going to use a unitary matrix from simulation then, maybe it's better to just use a statevector instead. There won't be any loss of phase either in that case. Is this a sound assumption? We need to run some experiments to validate this.
4. Do the relative phases of qubits in a circuit need to be calculated with real hardware (probabilistic)? Or can it be computed deterministically with a classical circuit,i.e., running a simulation is enough?

(*The following pages include detailed notes of our findings and experiments. This may not be a requirement for the submission; however, we added them in case they turns out to be helpful*)

Detailed Notes

Generating Unitary Matrix from Hardware and Synthesizing a New Circuit (Palvit)

Started by looking into getting unitary matrices from real hardware run, but kept getting the error "No unitary for experiment" as shown in screenshot below. Maybe it's not possible to get unitary matrices from real hardware run. We will discuss this with the professor in our next meeting.



Alternate approach tried:
Further, I tried to take the unitary matrix from the simulator run and recreated the circuit using transpile. The target was to see if we can achieve the same result count with a new circuit as the previous one using a simulator. The result doesn't come out to be the same. Seems like something is off with respect to hardware run, because the new circuit is generating correct counts with simulator run. The code is available in git repo here.

Methods for Generating Unitary Matrices from Simulation and Hardware (Harshwardhan)

Researched different methods to convert and store the unitary matrix generated in the hardware output for further use. Hardware simulation did not allow the formation of a unitary matrix, so I tried to run the circuit with a statevector simulator, which did not yield the desired results. The counts generated by the statevector simulator came out as empty.

Significance of Losing Phase Information because of Breakpoint Measurement (Shawn)

---

1. *What is in a unitary matrix? What does it contain?*
   For a quantum circuit with *n* qubits, the unitary matrix at any point is represented by a 2^n x 2^n matrix. This unitary matrix [contains the result of unitary transformations of the qubits up until that point](). We could construct an equivalent circuit based on some basis gates from that unitary matrix in qiskit. we can define equivalence of quantum circuits whenever their outputs are the same upon the same input.A unitary contains the probability distribution of a quantum circuit.

2. *What's the phase of qubit and why is it important?*
   "In quantum mechanics, particles have such a spin, too. It is the phase. In physical terms, the quantum mechanical phase originates from the concept that every quantum entity may be described not only as a particle but also as a wave."[1]. "...two waves [that] can have the same amplitude and the same wavelength, but they can differ in the relative positions of their crests and troughs. This **relative position is the phase** of the wave. Individually, the waves appear identical and they don't matter. However, when they interfere with each other, they do matter."

   Probability amplitude is unaffected by phase change. What effect does a phase change have on the overall result of a quantum circuit?

3. *What do we lose when you measure a quantum circuit and save it into a unitary matrix?*
   Lose phase information. Preserve the probability amplitudes.

4. *Would a quantum circuit be **incorrect** if you lose the phase information?*
   Perhaps. Without diving too much into details, a number of algorithms use quantum gates that involve phase changes, including Shor's algorithm.

5. *Can the local phases be determined probabilistically? Or can it be computed deterministically with a classical circuit,i.e., running a simulation is enough?* [**TODO]**

6. *In what way the reconstructed gate and original circuit differ? How much in terms of measurement?* [**TODO**]

7. *In case of multiple breakpoints, do the errors propagate due to the loss of phase?* If so, what's the deviation? **[TODO]**
8. *Does having certain gates on a given circuit is more prone to the error propagation issue?*
   Circuits consisting of phase gates such as Z, which only transforms the phase of a qubit, should be a good experiment specimen

---

[1]https://towardsdatascience.com/the-qubit-phase-b5fea2026ea#:~:text=The%20quantum%20bit%20(qubit)%20is,%E2%9F%A9%2B%CE%B2%7C1%E2%9F%A9.

9. *Do we need to worry about the global phase?*
   We cannot measure the global phase. And more importantly we do not have to worry about the global phase because it does not add anything meaningful to the circuits - only the relative/local is what's important.