

[Sign in to Platform](#)[Quantum Research](#)[↳ Blog](#)

How to measure and reset a qubit in the middle of a circuit execution

Date

11 Feb 2021

Authors

Paul Nation

Topics

Share this blog



[Sign in to Platform](#)

dynamic circuits, as highlighted in our recently released [Quantum Developer Roadmap](#). Dynamic circuits are those circuits that allow for a rich interplay between classical and quantum compute capabilities, all within the coherence time of the computation, and will be crucial for the development of error correction and thus fault tolerant quantum computation. However, there are many technical milestones along the way that track progress before we achieve this ultimate goal. Chief among these is the ability to measure and reset a qubit in the middle of a circuit execution, which we have now enabled across the fleet of [IBM Quantum](#) systems available via the IBM Cloud.

Measurement is at the very heart of quantum computing. Although often overlooked, high-fidelity measurements allow for classical systems (including us humans) to faithfully extract information from the realm in which quantum computers operate. Measurements typically take place at the end of a quantum circuit, allowing, with repeated executions, one to gather information about the final state of a quantum system in the form of a discrete probability distribution in the computational basis. However, there are distinct computational advantages to being able to measure a qubit in the middle of a computation.

Mid-circuit measurements play two primary roles in computations. First, they can be thought of as Boolean tests for a property of a quantum state before the final measurement takes place. For example, one can ask, mid-circuit, whether a register of qubits is in the plus or minus eigenstate of an operator formed by a tensor product of Pauli operators. Such “stabilizer” measurements form a core component of quantum error correction, signaling the presence of an error to be corrected. Likewise, mid-circuit measurements can be used to validate the state of a quantum computer in the presence of noise, allowing for post-selection of the final measurement outcomes based on the success of one or more sanity checks.

Measurements performed while a computation is in flight can have

[Sign in to Platform](#)

a highly entangled state, then a judicious choice of local measurements can “steer” a computation in a desired direction. For example, we can produce a three-qubit GHZ state and transform it into a Bell-state via an x-basis measurement on one of the three qubits; this would otherwise yield a mixed state if measured in the computational basis. More complex examples include cluster state computation, where the entire computation is imprinted onto the qubit’s state via a sequence of measurements.

Resetting a qubit

Closely related to mid-circuit measurements is the ability to reset a qubit to its ground state at any point in a computation. Many critical applications, such as solving linear systems of equations, make use of auxiliary qubits as working space during a computation. A calculation requires significantly fewer qubits if, once used, we can return a qubit to the ground state with high-fidelity. With system sizes in the range of 100 qubits, space is at a premium in today’s nascent quantum systems, and on-demand reset is necessary for enabling complex applications on near-term hardware. In Figure 1, below, we highlight an example of the quality of the reset operations on IBM Quantum’s current generation of Falcon processors, on the Montreal system, by looking at the error associated with one or more reset operations applied to a random single-qubit initial state.

Figure 1: we highlight an example of the quality of the reset operations on IBM Quantum’s current generation of Falcon_r4 processors by looking at the error associated with one or more reset operations applied to a random single-qubit initial state.

Internally, these reset instructions are composed of a mid-circuit measurement followed by an x-gate conditioned on the outcome of the measurement. These conditional reset operations therefore represent

[iterative phase estimation](#) algorithm. However, while the control techniques necessary for iterative phase estimation are still a research prototype, you can use mid-circuit measurement and conditional resets, today.

We can incorporate both concepts illustrated here into simple examples. First, Figure 2 shows a circuit utilizing both mid-circuit measurements and conditional reset instructions for post-selection and qubit reuse.

Figure 2: a circuit utilizing both mid-circuit measurements and conditional reset instructions for post-selection and qubit reuse.

This circuit first initializes all of the qubits into the ground state, and then prepares qubit 0 (q0) into an unknown state via the application of a random SU(2) unitary. Next, it projects q0 into the x-basis with eigenvalues 0 or 1 imprinted on q1 indicating if the qubit is left in the $|+\rangle$ (0) or $|-\rangle$ (1) x-basis states. We measure q1, and store the result for later use as a flag qubit for identifying which output states correspond to each eigenvalue. Step 3 of the circuit resets the already-measured q1 to the ground state, and then generates an entangled Bell pair between the two qubits. The Bell pair is either $|00\rangle+|11\rangle$ or $|00\rangle-|11\rangle$ depending on if q0 is in the $|+\rangle$ or $|-\rangle$ state prior to the CNOT gate, respectively. Finally, in order to distinguish these states, we use Hadamard gates to transform the state $|00\rangle-|11\rangle$ to $|01\rangle+|10\rangle$ before measuring.

Figure 3 shows the outcome of executing such a circuit on the seven-qubit IBM Quantum Casablanca system, where we see that the measurement of the flag qubit value measured before (in bold) correctly tracks the expected Bell states generated at the output. Collecting marginal counts over the flag qubit value indicates the proportion of the initial random q0 state that was in the $|+\rangle$ or $|-\rangle$ state after the projection. For the example considered here, these values are ~16 percent and ~84 percent, respectively. The dominant source of

[Sign in to Platform](#)

processor revisions will bring faster measurements, reducing the effect of this error.

Next, we consider the computational advantages of using reset to reduce the number of qubits needed in a 12-qubit Bernstein-Vazirani problem (Fig. 3). As written, this circuit cannot be implemented directly on an IBM Quantum system, but rather requires the introduction of SWAP gates in order to satisfy the limited connectivity in systems such as our heavy-hex based Falcon and Hummingbird processors. Indeed, compiling this circuit with Qiskit yields a circuit that requires 42 CNOT gates on a heavy-hex lattice. The fidelity of executing this compiled circuit on the IBM Quantum Kolkata system yields a disappointing 0.007; the output is essentially noise.

However, with the ability to measure and reset qubits mid-flight, we can transform any Bernstein-Vazirani circuit into a circuit over just two qubits requiring no additional SWAP gates. For the previous example, the corresponding circuit is:

And execution on the same system gives a vastly improved fidelity of 0.31; a 400x improvement over the standard implementation. This highlights how, with mid-circuit measurement and reset, it is possible to write compact algorithms with markedly higher fidelity than would otherwise be possible without these dynamic circuit building blocks.

Mid-circuit measurement and conditional reset represent an important first step toward dynamic circuits – and one that you can begin implementing into your quantum circuits as we speak. We’re excited to see what our users can do with this new functionality, while we continue to expand the variety of circuits that our devices can run. We hope you’ll follow along as we implement our development roadmap; we’re working to make the power of dynamic circuits a regular part of quantum computation in just a few years.

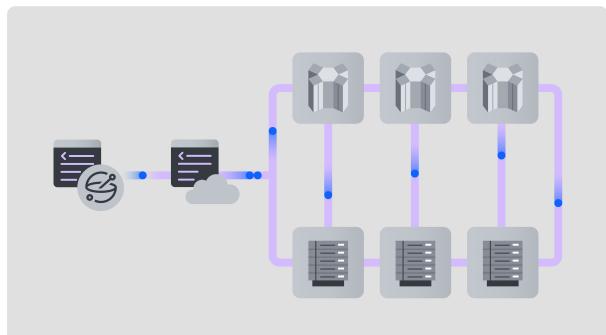
[Sign in to Platform](#)

IBM Quantum: Tomorrow's computing Keep exploring today

[View all blogs →
Quantum starts here](#)

Qiskit leads quantum software development kits in performance, testing shows

16 Sep 2024 • Paul Nation, David Kremer, Blake Johnson, Ryan Mandelbaum

[Qiskit](#)

Qiskit Serverless sets the stage for Qiskit Functions in the cloud

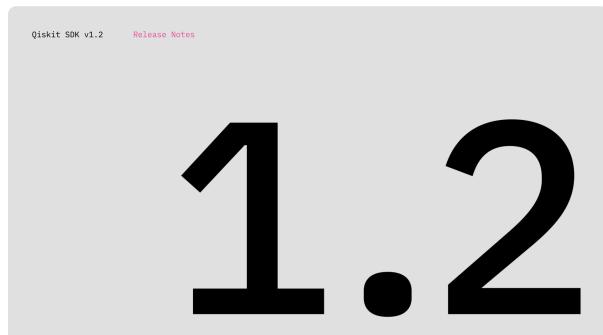
4 Sep 2024 • Kaelyn Ferris, Sanket Panda, Robert Davis

[Enablement](#)[Software](#)[Qiskit](#)

Qiskit Functions Catalog

Announcing the Qiskit Functions Catalog

16 Sep 2024 • Suhare Nur, Tushar Mittal, Jennifer Glick, Sanket Panda, Ryan Mandelbaum

[Software](#)[Qiskit](#)

Release news: Qiskit SDK v1.2 is here!

21 Aug 2024 • Qiskit Team

[Qiskit](#)

[Sign in to Platform](#)

IBM Quantum

Featured

- [Technology](#)
- [IBM Quantum Safe](#)
- [IBM Quantum Network](#)
- [Research](#)

Get access

- [Pricing](#)
- [IBM Quantum Platform](#)

Get started

- [Qiskit](#)
- [Documentation](#)
- [Learning](#)

Stay connected

- [Blog](#)
- [LinkedIn](#)
- [YouTube](#)
- [Community](#)
- [Careers](#)

[Contact IBM](#)[Terms of use](#)[Accessibility](#)[Privacy](#)[Cookie preferences](#)