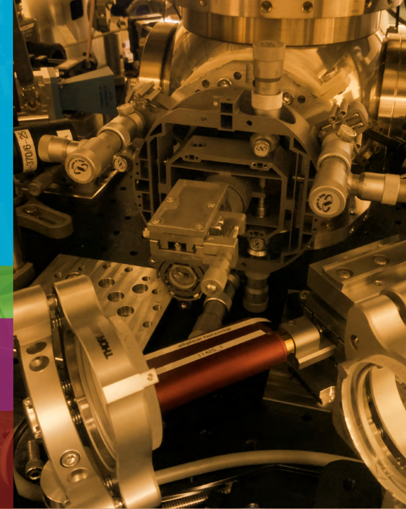
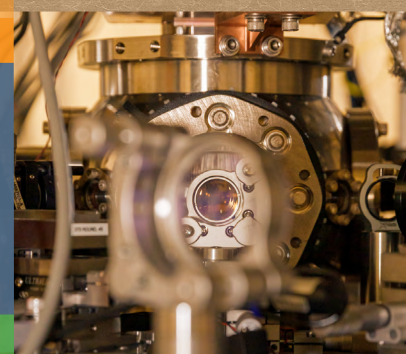
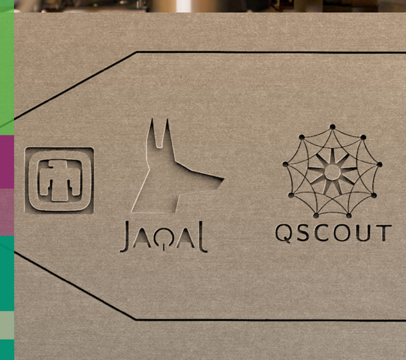
QSCOUT

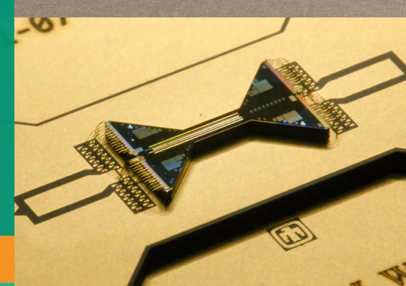
PROGRESS REPORT

June 2022









SANDIA NATIONAL LABORATORIES

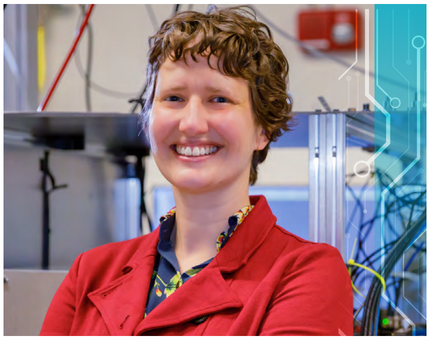
TABLE OF CONTENTS

MISSION TEAM HISTORY

Quantum information processing has reached an inflection point, transitioning from proof-of-principle scientific experiments to small, noisy quantum processors. To accelerate this process and eventually move to fault-tolerant quantum computing, it is necessary to provide the scientific community with access to whitebox testbed systems. The Quantum Scientific Computing Open User Testbed (QSCOUT) provides scientists unique access to an innovative system to help advance quantum computing science.

When the DOE Office of Science ASCR testbed program began in 2019, QSCOUT was one of two testbeds selected to carry out the DOE mission of providing collaborative facilities that allow external researchers to access novel, early-stage quantum computing resources. QSCOUT is based on trapped-ion technology, which is one of the most mature quantum computing platforms. In the three years of this project, we have transformed an empty lab to a quantum computing machine that is running user code written in our own quantum programming language Jaqal (Just Another Quantum Assembly Language). This feat required coordination from many teams within Sandia as well as two university partners, Tufts University and Duke University.

The QSCOUT team is broad and far reaching within Sandia:



Susan M. Clark

QSCOUT Principal Investigator

Dr. Clark is Principal Investigator of the QSCOUT project and a Principal Member of Technical Staff at Sandia National Labs. She has worked with a variety of quantum information platforms, including trapped ions, gate-defined quantum dots in silicon, and neutral donors in GaAs and ZnSe, with an emphasis on high-fidelity operations and quantum systems engineering. She received a Ph.D. in Applied Physics from Stanford University with Professor Yoshihisa Yamamoto and was awarded a Joint Quantum Institute Postdoctoral Fellowship at the University of Maryland, where she worked with Professor Christopher Monroe.

Rick Muller

QSCOUT Project Manager

Dr. Muller is the Senior Manager of the Quantum and Advanced Microsystem group at Sandia and manages the Sandia portfolio in Quantum Information Science. Rick is also Deputy Director of the Quantum Systems Accelerator, one of the five DOE National Quantum Information Science Research Centers, which is co-led by Lawrence Berkeley National Laboratory and Sandia, and includes collaborators at Harvard, MIT, Caltech, Duke, Berkeley, and other institutions.

Andrew Landahl

Theory and Software Team Lead

Dr. Landahl is a Distinguished R&D Scientist in the Quantum Computer Science department at Sandia National Laboratories, a Research Professor of Physics and Astronomy at the University of New Mexico, and a Fellow of the American Physical Society. Andrew came to Sandia in 2009 after serving as a full-time Research Assistant Professor at UNM. Prior to that, he was a Hewlett-Packard Postdoctoral Fellow at MIT with extended visits to Cambridge University and Harvard. He holds Ph.D. and M.S. degrees in Physics from Caltech and B.S. degrees in Physics and Mathematics from Virginia Tech.

Christopher G. Yale

Experimental Team Lead

Dr. Yale leads the experimental effort for the QSCOUT project and is a Senior Member of the Technical Staff at Sandia National Labs. His current research focuses on trapped-ion quantum computing, and has previously worked with other qubit implementations, including optically-controlled nitrogen-vacancy centers in diamond and ultracold polar molecules. He received a Ph.D. in Physics from University of California, Santa Barbara doing research both there and at University of Chicago with Professor David Awschalom, and did postdoctoral work at University of Chicago and Sandia National Labs.

Melissa C. Revelle

Optical Engineering Lead and Trap Fabrication Liaison

Dr. Revelle is a Principal Member of Technical Staff at Sandia National Labs, PI of the IARPA LogiQ program, and a lead member of the QSCOUT team. Her work has covered many fields in atomic physics including degenerate Fermi gases, atomic interferometers, Bose-Einstein condensates, and trapped-ions. Through these, she has focused on high precision optical systems and integration into quantum systems. Melissa came to Sandia as a postdoctoral fellow after earning her Ph.D. in Physics in 2016 from Rice University in Texas.

Daniel Lobser

Hardware Implementation Team Lead

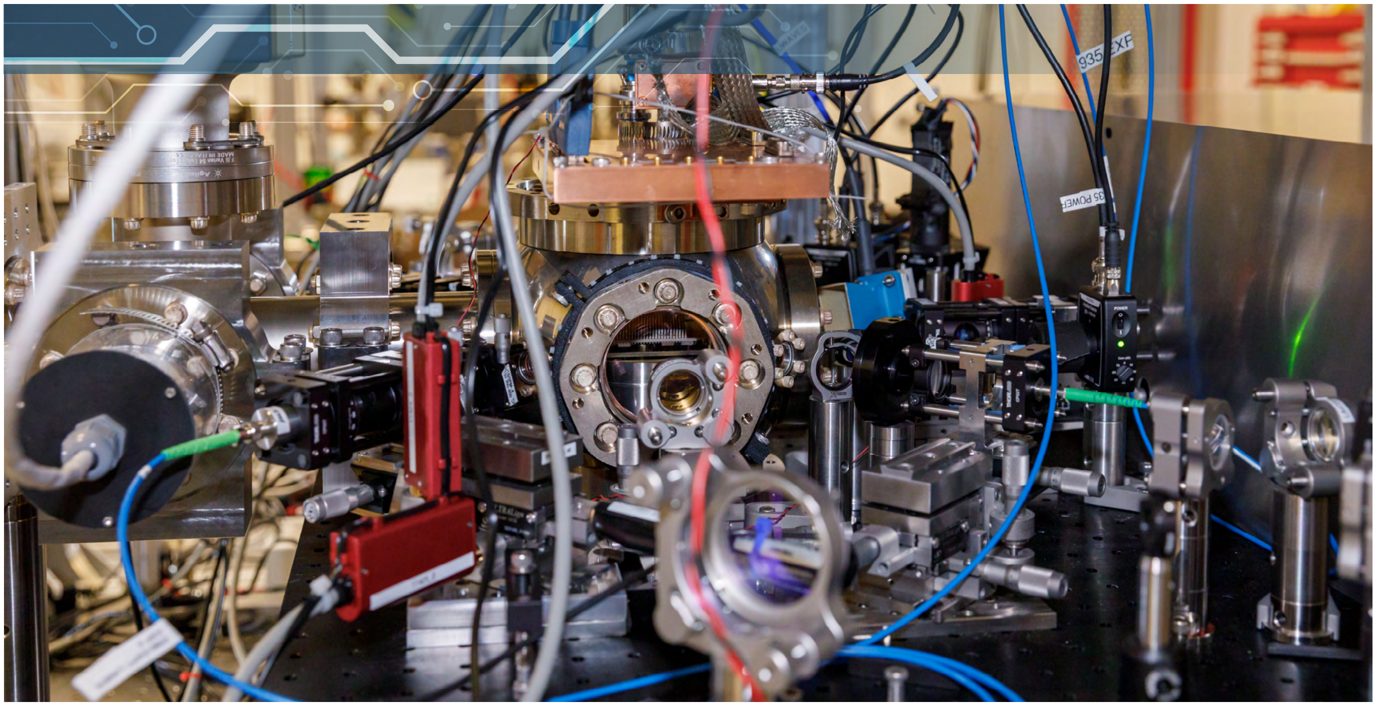
Dr. Lobser leads the control systems thrust of QSCOUT. His main research interest is the development of custom classical and quantum control hardware that employs novel paradigms for coherent operations and dynamic noise mitigation in trapped-ion qubit platforms. He received his Ph.D. in physics from the University of Colorado, Boulder in 2015, where he studied ultracold atomic gases.

"As a mechanical engineer, quantum computing is far outside my wheelhouse but has been a very rewarding field. Requirements change quickly and sometimes come at the last minute. There are many unique challenges that require creative solutions and I feel like I've really been able to use my expertise in materials and mechanical design to address those challenges and advance the QSCOUT project."

Bradley Salzbrenner
Mechanical Team Lead

- **Experimental Team**: Ashlyn Burch, Matthew Chow, Craig Hogle, Megan Ivory, Daniel Lobser, Theala Redhouse, Melissa Revelle, Joshua Wilson, Christopher Yale
- **Theory and Software Team**: Benjamin Morrison, Kenneth Rudinger, Antonio Russo, Brandon Ruzic, Kevin Young
- **Software and Hardware Implementation Team**: Joshua Goldberg, Daniel Lobser, Jay Van Der Wall
- **Mechanical Team**: Madelyn Kosednar, Joshua Lane, Bradley Salzbrenner, Edward (Ted) Winrow
- **Fabrication Team**: Matthew Delaney, Edwin Heller, Nick Jimenez, Becky Loviza, Zach Meinelt, Eric Ou, John Rembetski
- **Packaging Team**: Raymond Halti, Tipphachane Jennings
- **Tufts University**: Peter Love, Oliver Maupin, William Simon
- **Duke University**: Kenneth Brown

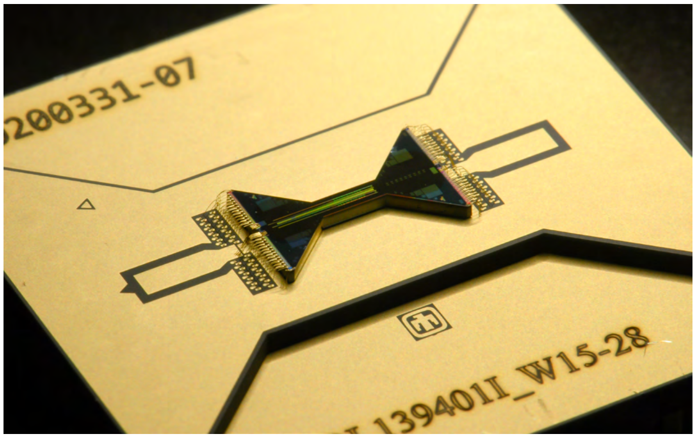
ARCHITECTURE NOTABLE ADVANCEMENTS

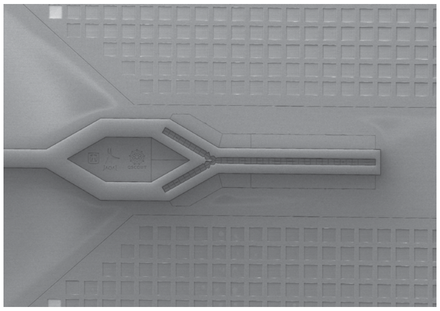


The QSCOUT platform is based on trapped ions, with many of the details outlined in [Clark *et al*. 2021]. For the initial hardware iteration, all ions are trapped in a single trapping zone. Each ion is addressed with a dedicated qubit laser beam, and each ion is aligned to a core in a multicore fiber that goes to a dedicated photomultiplier tube (PMT) for distinguishable detection. This architecture enables all-to-all connectivity between ions and individual control to large numbers of ions.

TRAP DESIGN

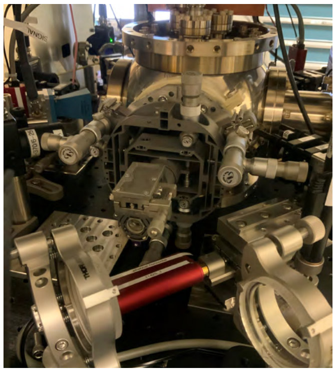
The QSCOUT program supports the development of three trap fabrication efforts. The first QSCOUT trap is the Peregrine, a multi-level metal surface design, as outlined in [Revelle 2020]. This is a linear ion trap, in which there are no junctions and ions can only be moved along the length of the trap. This trap is intended to be a straightforward workhorse trap and serve the needs of first- and second-round users by providing a stable design capable of aligning ions to the laser beams and the detection system.





The second trap is Y-shaped and supports an architecture that will allow users to test ideas in error correction, which is important for the future of fault-tolerant quantum computing. This involves moving data ions out of the detection region to protect them while the other ions are detected. Then, depending on the result from the detection, the computation can resume with the protected data ions. This trap has just finished its fabrication process and testing will begin soon.

The third trap is is in development; its objective is to incorporate multiple trapping zones that allow users to study and compare different schemes of connectivity. While all the ions in the same zone will be fully connected, connections between zones may have an impact on algorithmic performance. This trap will ultimately allow us to push the frontiers of trap development as well as increase the available functionality.



QUBIT LASER SYSTEM

To control the internal state of one ion while negligibly impacting the neighbor, we created an optical system that is stable at the nanometer level and focused to a spot size much less than the spacing of the ions. Either of these challenges alone are difficult to overcome. QSCOUT directed the design of a system with mechanical control and stability that is indexed to the vacuum chamber and custom built for our optics. This 5-axis flexure stage is 3D printed from steel and titanium, resulting in a structure that is light and stiff and thus capable of the nanometer control needed to align the laser to the ions.

QSCOUT DEFAULT PARAMETERIZED GATESET

QSCOUT provides a unique gateset to our users in contrast to most commercial quantum-computing offerings. In particular, QSCOUT provides what are known as parameterized single- and two-qubit gates. A qubit, the basic computing unit of a quantum computer, unlike a classical bit, not only exists as 0 or 1, but can exist in any superposition of those two states along a surface known as the Bloch sphere. Likewise, when we scale to two qubits, their entangled states exist on a multi-dimensional hypersphere. Gates are the method by which the qubit state traverses these spheres as rotations about different axes.

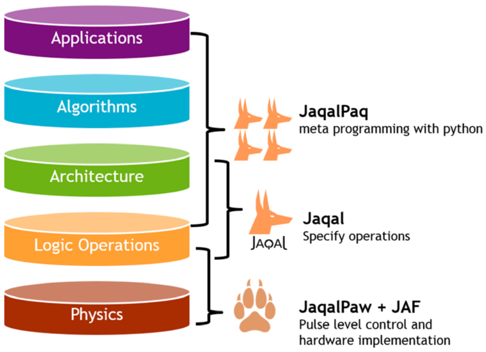
In the case of QSCOUT, we offer full parameterization of single-qubit gates. These gates can be about any axis on the Bloch sphere and rotated by any angle about that axis. Rotations in the XY (equatorial) plane are achieved by Raman pulses driven by lasers, while rotations about the Z-axis are done virtually via a phase advance.

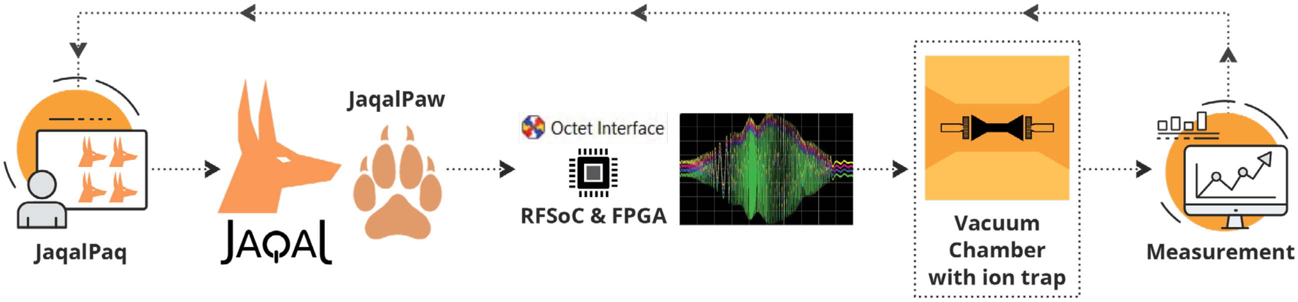
More uniquely, we also offer significant parameterization of the two-qubit gates in QSCOUT. The natural two-qubit gate in an ion-trap system is known as a Mølmer-Sørensen (MS) gate, which is an XX-type interaction on the Bloch hypersphere. A standard trapped ion gate set, such as those used by the commercial testbeds at IonQ or Quantinuum, provide an XX or ZZ interaction with a fixed rotation angle of $\pi/2$. For QSCOUT, we expand that offering to allow our users to select a parameterized MS gate, meaning they have the ability to define both the phase and the rotation angle of that interaction. In doing so, we provide a fuller set of gates to more efficiently achieve their desired algorithms. These parameterized two-qubit gates were a key feature that attracted half of our round one users. Throughout the first round, we've improved our techniques to achieve these gates, and plan to offer even more customizability and parameterization as QSCOUT continues.

SOFTWARE STACK

The software architecture is a critical piece of QSCOUT, and was developed from the ground up. This gives QSCOUT users a level of control and flexibility not possible with any production-level software environment for quantum computers. Our software stack includes the following:

- Python metaprogramming for easier programmability by testbed users (JaqalPaq)
- A quantum assembly language with full control of quantum-gate scheduling. (Jaqal: Just Another Quantum Assembly Language)
- An efficient and precise mechanism for specifying pulses and waveforms that effect quantum gates. (JaqalPaw: Jaqal Pulses And Waveforms)
- An application framework for executing quantum assembly experiments on hardware. (JAF: Jaqal Application Framework)





QSCOUT

- User builds JaqalPaq to build Jaqal circuits
- Jaqal circuits may use JaqalPaw to define new gates
- These are interpreted by Octet, the RFSoC Firmware, and transformed into laser pulses
- Laser pulses are sent to ios, and a quantum circuit is performed
- Measurement results are returned to users via the Jaqal Application Framework

JAQAL

The Jaqal programming language fills many gaps in existing quantum assembly languages that were absent when QSCOUT began. We presented a full formal specification of the language in [Landahl, et al. (2020)]; for a full discussion of the motivations behind the development of Jaqal, see [Morrison, et al. (2021)]. We summarize some of these motivations and features here.

- Jaqal qubit allocation binds specific hardware qubits to specific names. Many commercial languages, especially high-level ones, obscure this, giving the quantum hardware the option of serving up only the "best" qubits in an allocation instruction. OpenQASM and Cirq are notable exceptions, which do allow such bindings.
- Jaqal requires parallel and serial gate scheduling to be explicit, so they occur as directed by the testbed programmer. Many commercial languages leave this unspecified, or at best use "barrier" statements for synchronization. This leads to unknown quantum hardware behaviors to quantum assembly programmers, which is counter to the philosophy of testbed programming. Cirq is a notable exception, with limited parallel/serial scheduling constructions. However, even Cirq cannot express a serial sequence of gates on one qubit scheduled in parallel with a (longer duration) gate on another qubit, like Jaqal is able to do.
- Default pulse-and-waveform (JaqalPaw) programs exist for every native Jaqal gate, and users can specify new Jaqal gates at the JaqalPaw level. Even users who do not want to innovate their own gates can still view the internals of Jaqal gates in JaqalPaw for debugging purposes. When we developed this capability, OpenQASM did not allow for the definition of new native gates via OpenPulse, although this appears to be changing.
- Jaqal offers fixed-iteration looping constructs, which avoids the possibility of non-halting programs, while offering instinctive classical control flow. Languages like Q#, with repeat-until and while loops, and Quil, with jump instructions, run the risk of non-halting quantum programs.

JAQALPAQ

At a layer of abstraction above the Jaqal language, we created a classical metaprogramming environment in Python that we call JaqalPaq. For all the details and supporting documentation of JaqalPaq, as well as the option to download and use this open-source package yourself, please go to https://gitlab.com/Jaqal. JaqalPaq treats Jaqal programs as objects in an object-oriented paradigm. By pushing the classical control to the python level, QSCOUT avoids having to re-invent any classical programming constructs—Jaqal remains a purely quantum language. JaqalPaq not only offers a metaprogramming environment, it also offers many other features helpful to testbed developers including the following:

- Noise-free (unitary) emulation of any Jaqal program, with results returned as quantum state vectors or sampled classical data from it.
- Noisy Markovian (process-matrix) emulation of any Jaqal program, with the ability to define one's own noise model per gate. Results can be returned as density matrices or sampled classical data.
- Built-in Markovian noise models developed by Sandia experts on the QSCOUT hardware, which testbed developers can simply use without having to generate their own noise models. These noise models are constructed at the JaqalPaw and atomic physics level and pushed up to effective process matrices at the Jaqal level. This is a good option for users who want to predict what will happen with their programs before they run them on actual QSCOUT hardware.

- Transpilation tools that allow one to convert an existing quantum assembly program from another language into Jaqal. Because no other language is as flexible or precise at scheduling at Jaqal is, our transpilers necessarily make *ad hoc* guesses as to what the program's scheduling intent. They also convert gates to QSCOUT's native gate set in a way that is not necessarily optimal. For this reason, the transpilers are a good way to get started for new users, but they are unlikely to generate the best possible quantum assembly code that represents the intent of the testbed programmer.

JAQALPAW

Our JaqalPaw suite is tailored to the trapped-ion hardware platform upon which QSCOUT is built, but it incorporates an extensible design philosophy that allows JaqalPaw code to be ported to other platforms in which specific or highly-customized hardware systems must be targeted. Additionally, the scalable coherent control system developed for QSCOUT, known as Octet, is currently being used in numerous labs outside of Sandia—this is in no small part due to its wide array of features for generating arbitrarily modulated and synchronous rf waveforms needed to realize state-of-the-art quantum gate designs—for multi-qubit quantum information systems.

JAQAL APPLICATION FRAMEWORK

The Jaqal Application Framework (JAF) is the mechanism by which jobs are submitted to the quantum hardware. It is a client-server architecture running over our in-house Icypaw protocol. The server runs in its own Docker container on a Linux machine, and communicates with IonControl via a custom graphical user interface (GUI).

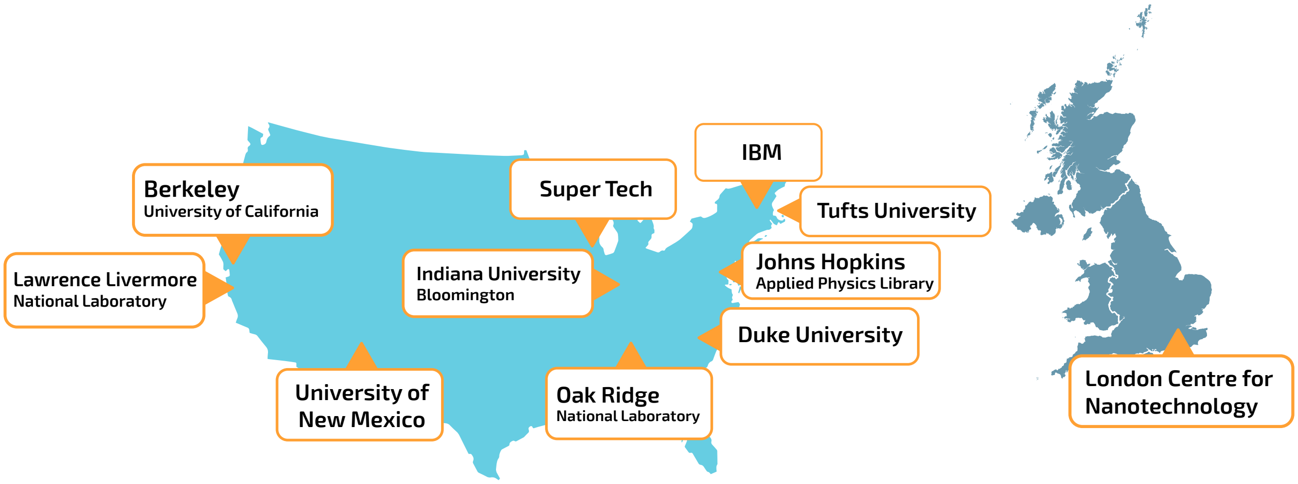
The division of the JAF into client and server serves two purposes. First, it eases software maintenance by separating the logical execution of quantum programs from their physical realization. Second, it allows us to create a custom environment with its own Python packages for executing quantum programs provided by our users. JAF supports running Jaqal circuits directly or executing a Jupyter notebook that iteratively produces Jaqal circuits to be executed on the quantum hardware. The software framework has been set up so that users may create a notebook which will run on the emulator or the hardware without modification.

RFSOC ARCHITECTURE

To realize the features available in the software on the QSCOUT system, we upgraded the control hardware to use an RF System on Chip (RFSoC). By fully utilizing the device's programmable logic, quad-core ARM processor, and real-time ARM processor, we can implement an asymmetric, multi-processing architecture that offers a familiar high-level interface with a Linux operating system, fast and deterministically timed algorithms for experimental control flow, and highly customized peripherals in field programmable gate array (FPGA) fabric. Embracing the modern architecture in these devices circumvents issues associated with custom CPU designs implemented in an FPGA that hurt performance and waste valuable resources, allowing us to fully utilize programmable logic for curated peripherals that are specially tailored to our experimental system, while maintaining the benefits of a flexible classical computing system. The real-time processors allow multiple boards to be used as a composite system, providing a control system with scalability. We've developed initial firmware designs, device drivers, and a real-time processor application progra interface (API) to fill out the rest of our hardware toolbox. This design is compatible with Xilinx multi-processing system-on-chip (MPSoC) and radiofrequency system-on-chip (RFSoC) architectures to support larger amounts of general-purpose I/O, cost effectiveness, and uniformity across hardware subsystems.

TESTBED USER PROGRAM

The QSCOUT proposal selection process begins with a call that is advertised on the QSCOUT web page and in other forums where interested researchers will see it (conference presentations, emails to past submitters, etc). The call describes the anticipated performance of the QSCOUT system, technical support available to selected users, and a description of the two-page proposal format. Applicants describe the scientific questions to be addressed, the current state of research, why the QSCOUT testbed is suitable for the proposed work, and the expected impact. QSCOUT accepts proposals from anywhere in the world.



About 15 proposals were submitted by researchers in each of the first two rounds, representing universities, national labs, and companies. A five-person panel consisting of a Sandia chairperson, two university QSCOUT contributors, and two rotating positions filled by non-Sandia government lab staff ranked the proposals according to technical merit, feasibility, and impact. The panel ultimately selected five to six proposals for each round. The awardees were identified and feedback was provided to those who were not selected. Several groups submitted proposals in both rounds.

Each proposal winner is promised up to 80 hours of run time on the machine free of charge or until their dataset is complete (whichever comes first). Calibration and maintenance time is not counted against user's teams' time. The price of maintaining and running the machine is paid for by the DOE Office of Science.

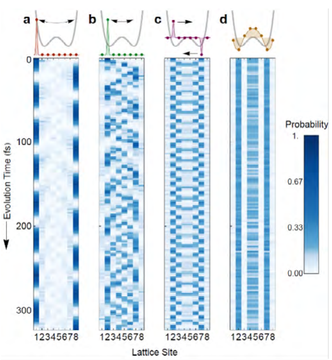
2021 R&D 100 WINNER

In 2021, QSCOUT won an R&D 100 award for its unique access to low level quantum hardware.

SCIENTIFIC HIGHLIGHTS

Below is a summary of research performed by our first round users. One of the key features of the QSCOUT program's collaborative interface with our users is the co-design and co-development of features on QSCOUT. In particular, during interactions with each of our user teams, we have made significant and important design changes and developments based on our users' projects needs

Indiana University Team: Philip Richerme, Debadrita Saha, Sam Norrel, Jeremy Smith, Amr Sabry, Srinivasan Iyengar



***Quantum Computation of Hydrogen Bond Dynamics and Vibrational Spectra;* arXiv: 2204.08571**

Simulating Proton-coupled electron transfer problems.

Solving the electron-nuclear wavefunction within molecules is a grand challenge problem in quantum chemistry. In such systems, coupled proton-electron degrees of freedom may behave in a correlated fashion and as the number of these degrees of freedom grow, the complexity of the problem quickly renders itself classically intractable. This project proposed using a mapping from Ising-type Hamiltonians (ions) to a proton coupled electron transfer (PCET) Hamiltonian, thus enabling the QSCOUT system to experimentally simulate the quantum dynamics of PCET problems in quantum chemistry. They demonstrated a unique mapping of the nuclear lattice Hamiltonian to an Ising Hamiltonian and simulated it with the QSCOUT machine. Through the experimental data, the energy level diagram of an exemplar molecule (DMANH+) was reconstructed to within spectroscopic accuracy.

NEW CALIBRATION ROUTINES

Our collaborators at Indiana University requested an arbitrary two-qubit unitary interaction which could be decomposed into 3 controlled-NOT (CNOT) gates. The CNOT is a classic gate in quantum computing, but it is not a natural gate within a trapped ion system. As such, the CNOT needs to be decomposed into a series of single-qubit gates and one two-qubit MS gate. As Indiana's code was the first user code we ran after developing two-qubit gates, we found the CNOT composite gate acted as an efficient way to determine and calibrate the appropriate phase relation between our single- and two-qubit gates.

Oak Ridge National Laboratory Team: Swarnadeep Majumder, Titus Morris, and Raphael Pooser

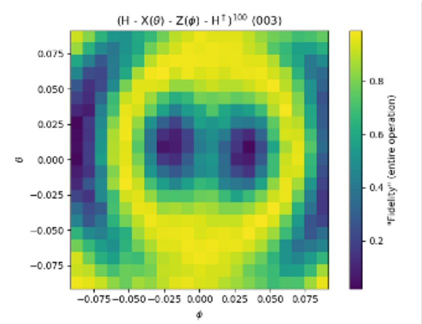
Characterizing and mitigating coherent errors in a trapped ion quantum processor using hidden inverses.

The goal of this project was to characterize and mitigate coherent errors through the use of hidden inverses. Here, we examined how the single-qubit Hadamard and its inverse as well as the two-qubit CNOT and its inverse could be used to characterize the types of noise present in the system. Additionally, by intentionally introducing errors, we were able to asses the performance of this error mitigation technique in the presence of certain types of noise.

BATCHING FOR EFFICIENT USE OF RESOURCES

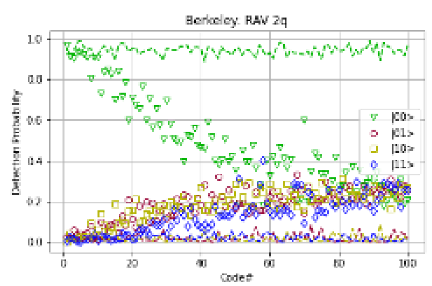
Oak Ridge National Laboratory requested a technique called randomized compiling, which consists of including extra random gates to see their effect on noise processes. However, to implement this idea, a new Jaqal file needed to be generated for each randomization, which was inefficient and slow. To address this, the software and hardware implementation team modified the architecture to allow for batching, which allows the user to provide multiple Jaqal files as a batch to the Octet system, instead of Octet waiting for a new code to be communicated and uploaded for each new file. While all users benefited, ORNL benefited the most. In their case, batching reduced 3690 different Jaqal code transfers and uploads to 41, greatly decreasing the time it took for their code to run.

In particular, we found we could use the Hadamard and its inverse in a sequence with small-angle single-qubit gates to extract details about calibration error. Likewise, by using the CNOT and its inverse in a variational quantum eigensolver (VQE), we demonstrated a robustness to intentionally introduced coherent and incoherent error.



Characterizing and mitigating coherent errors in a trapped ion quantum processor using hidden inverses; *arXiv:2205.14225*

University of California, Berkeley Team: Ryan Shaffer, Hang Ren, Emiliia Dyrenkova, Hartmut Häffner
Efficient verification of continuously-parameterized quantum gates



Efficient verification of continuously-parameterized quantum gates; *arXiv:2205.13074*

In this project, we investigated a benchmarking technique known as random analog verification (RAV), an approach ideally suited to the QSCOUT system as it relies on using a fully parameterized single- and two-qubit gateset. Essentially, RAV is a random walk across the two-qubit Bloch hypersphere followed by an approximate inverted sequence to return the qubits close to their initial state. It is directly compared to cross-entropy benchmarking (XEB), which is just the first random walk with no inversion sequence. When compared, fidelity estimations made via RAV have 2-5 times less variance than those made via XEB given the same number of experiments. To get to similar precision on the fideilty estimates, XEB requires 6-25 times more experiments. Both RAV and XEB provide similar error rate estimations for QSCOUT, an error per circuit layer of 1.4 x 10^-2.

INCREASING CIRCUIT DEPTH

The user team at University of California, Berkeley has been studying different benchmarking techniques, including a homegrown benchmarking techniques, random analog verification (RAV). As the RAV benchmarking approach relies on the ability to generate a random set of parameterized single- and two-qubit gates with total circuit depths approaching 1000 circuits, it provided both a benchmark and a way to stress-test our control hardware. Unlike most approaches where long circuits use only a handful of unique gates, here each gate is a truly unique instantiation, based on its parameterization. Above circuit depths of 300 unique gates, we realized our control hardware, Octet, was unable to process the remainder of the pulse data objects. In response, we increased the size of Octet's buffer four-fold as well as included the ability for partial reprogramming for especially long parameterized sequences.

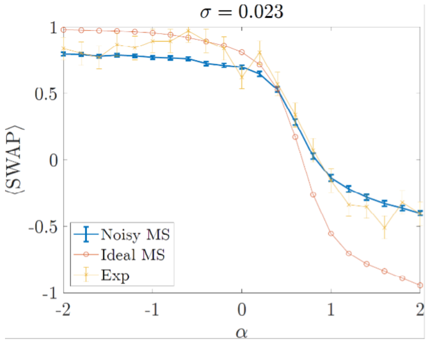
CROSS-POLLINATION BETWEEN USER TEAMS

There are a number of interdependencies between our user collaboration teams. In many instances, we will find a technique or approach works well for one user team and, with that team's permission, suggest that approach to another user team. Also, improvements to calibration techniques (especially in the case of our parameterized MS gate) are automatically implemented for all users.

Our users also recently participated in our first-ever user summit in March, which resulted in beneficial cross-pollination of ideas. Due to the depth and sensitivity to noise of our UNM team's code, they're starting to investigate noise mitigation techniques, partially inspired by the work we've done with our ORNL team.

University of New Mexico Team: Tameem Albash, Namitha Pradeep, Elizabeth Crosson, Milad Marvian
Digital simulation of adiabatic evolution

This research aims to study a digitized simulation of an adiabatic evolution, which remains an important algorithm for ground state preparation. The interpolating Hamiltonian has enough flexibility to provide a wide range of adiabatic conditions, making it a suitable benchmark. The objective is to use this flexibility to improve practical implementations of digitized simulations as well as to be a sensitive probe of the noise on the QSCOUT hardware.

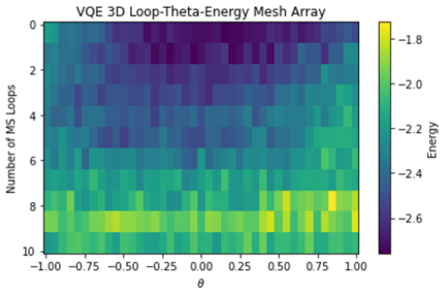


In particular, given the current capabilities in the system, we needed to modify the initial proposed circuits to reduce the total number of two-qubit gates through the introduction of a counter-diabatic term to the simulations. Initial results show some agreement with the simulations, but understanding the impact of system noise on the simulation is largely unclear. Efforts are underway to improve performance through the use of randomized compiling and dynamical decoupling to better elucidate the effects of noise on the digitized simulation.

PARAMETRIZED TWO-QUBIT GATES

The University of New Mexico (UNM) is reliant on the use of parameterized two-qubit gates to generate their arbitrary two-qubit unitary interaction. Through the analysis of the performance of their code on the system, we are able to improve the performance and calibration of our two-qubit gate. With their collaboration, we are examining many of the finer details pertaining to issues surrounding the use of these gates, including how AC Stark shifts affect gates of different angles, an angular dependence of unwanted leakage out of qubit subspaces, and the effects of stochastic noise processes on the gate itself.

Tufts University: Oliver Maupin, Peter Love
Exploring extrapolation techniques to reduce the error in VQE algorithms run using Jaqal (University partner)

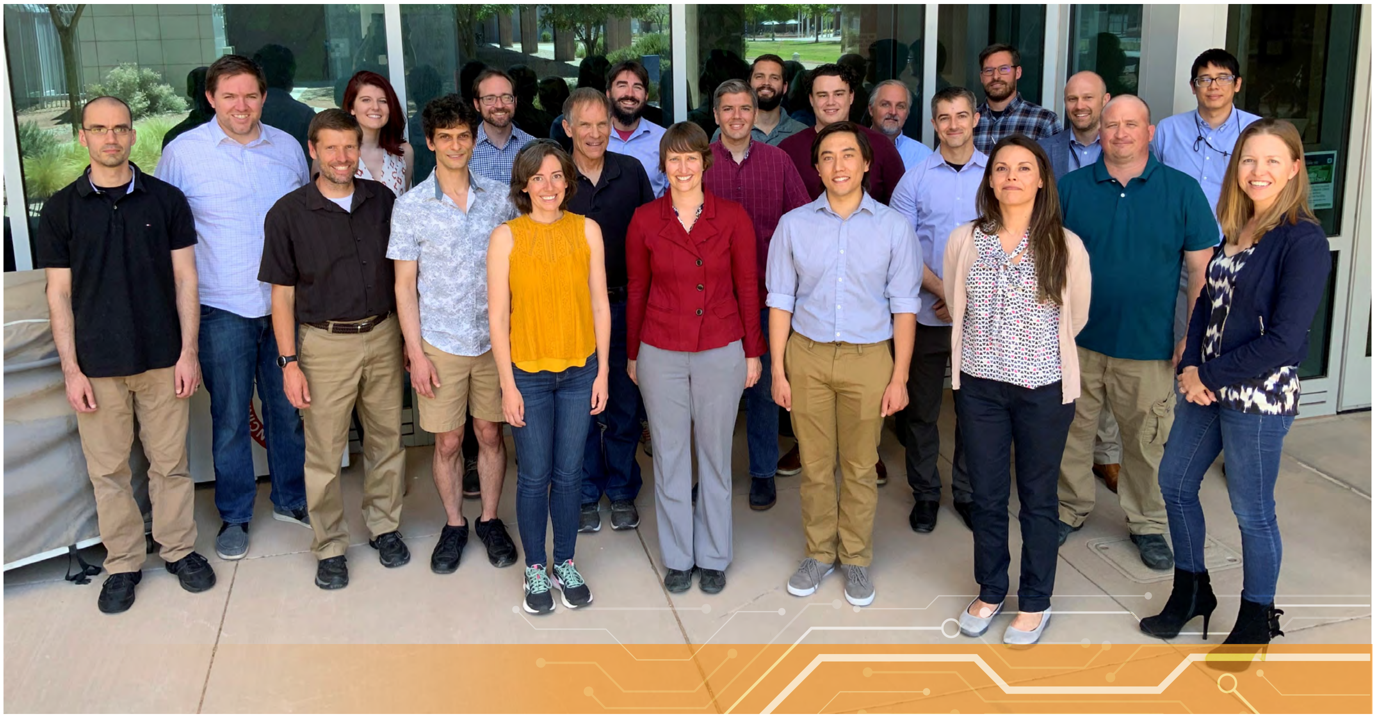


In this research, we performed a VQE algorithm on the QSCOUT hardware to determine the ground state energy of a Hamiltonian and used extrapolation techniques to reduce the level of calculation error due to noise in the system. Since a traditional Richardson time stretch was not providing enough variation in the data to extrapolate to minimal error, a unitary folding extrapolation technique was found to better match the experimental data. In this regime, the noise was increased in the circuit by adding more gates and an appropriate scaling factor was determined to match with the level of folding. After performing the VQE algorithm on QSCOUT, optimized energies were calculated to be within a few percent error from the actual FCI energy. The second and third order extrapolations over the optimized energies revealed an even closer comparison to the ground state energy.

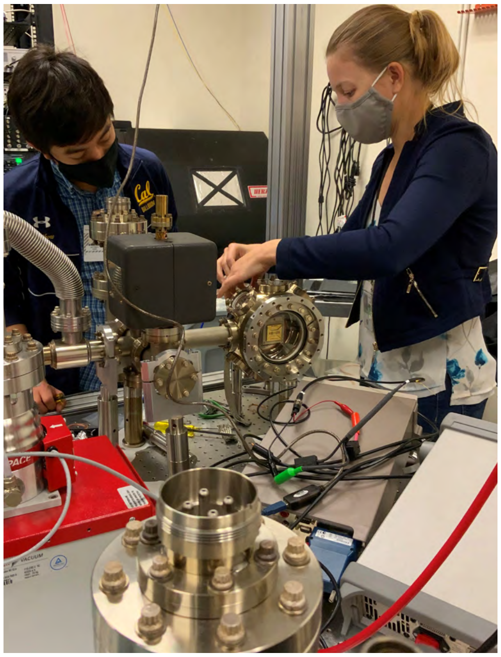
EXEMPLAR CODES

Not a user team, but our collaborators at Tufts have worked with us to develop a series of Jaqal exemplar programs based on variational quantum eigensolvers (VQE). As part of this collaboration, the software interface to run user code, the Jaqal Application Framework, was developed in tandem with and based on these exemplar programs. Likewise, the VQE exemplar programs were helpful to the ORNL team as they developed their VQE instantiation using hidden inverses.

BUILDING THE QUANTUM WORKFORCE



Given their openness, the testbeds are a natural place to educate the next generation of quantum system engineers and operators. This past year, QSCOUT has served as an educational tool for students, postdocs, and staff members to learn the details of quantum system design and general skills related to ion-trap quantum computing.



Our postdocs and students are given the opportunity to work on all aspects of the system. This includes working to develop the hardware from the ground-up, interacting with the software to make each experiment more compatible to each user, and writing software. Some of the postdocs were assigned to act as points of contact with the users and work directly with them to help run their experiments.

A number of other team members at Sandia come from quantum-adjacent fields, including mechanical engineers, electrical engineers, optical engineers, fabrication engineers, and software engineers. These workers are included in weekly meetings to facilitate exposure to the bigger picture of QSCOUT's objectives and how their contributions fit in.

Beyond the workforce at Sandia, the operation of QSCOUT as a user testbed provides the opportunity for academic and industry partners to have access to a state-of-the-art quantum computer with low-level access that they otherwise might not have. Several user proposals were written and led by postdocs and graduate students, giving them valuable leadership experience and extending the workforce development and educational aspects beyond those at Sandia alone.

RESOURCES

- QSCOUT Website: https://qscout.sandia.gov

REFERENCES

- Susan M. Clark, D. Lobser, M. Revelle, C. Yale, D. Bossert, A. D. Burch, M. N. Chow, C. W. Hogle, M. Ivory, J. Pehr, B. Salzbrenner, D. Stick, W. Sweatt, J. M. Wilson, E. Winrow, and P. Maunz. (2021) "Engineering the Quantum Scientific Computing Open User Testbed." *IEEE Transactions on Quantum Engineering* vol. 2, pp. 1-32 Art no. 3102832, doi: 10.1109/TQE.2021.3096480.
- JaqalPaq https://gitlab.com/jaqal/jaqalpaq
- Andrew J. Landahl, Daniel S. Lobser, Benjamin C. A. Morrison, Kenneth M. Rudinger, Antonio E. Russo, Jay W. Van Der Wall, and Peter Maunz (2020). "*Jaqal, the Quantum Assembly Language for QSCOUT*." arXiv:2003.09382 [quant-ph].
- B. C. A. Morrison et al. (2020), "Just Another Quantum Assembly Language (Jaqal)," *2020 IEEE International Conference on Quantum Computing and Engineering* (QCE), pp. 402-408, doi: 10.1109/QCE49297.2020.00056.
- M. C. Revelle "Phoenix and Peregrine Ion Traps," arXiv:2009.02398 (sept. 2020)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration