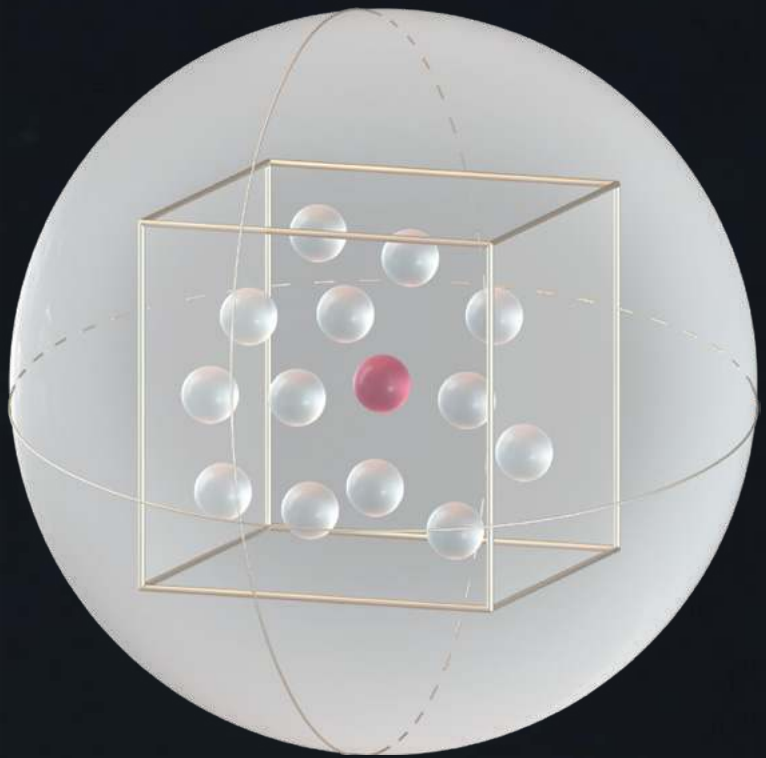
QUANTUM MACHINES

USE CASE

Quantum Orchestration for Quantum Dots

With the Quantum Orchestration Platform, there's no limit to the kind of experiments you can run.
Find out how to supercharge your quantum dots research with these real-world use cases.



INTRODUCTION TO QUANTUM DOTS

In quantum dot (QD) – based qubits, charge and spin are controlled in precise ways in a solid-state device. There are various ways to create suitable quantum dots, define qubits from them, and different methods to control them and readout their state [Floris A. Zwanenbur et al., Silicon quantum electronics. Reviews Of Modern Physics, 85(3), 961-1019 (2013)]. As a concrete example, we consider the system developed in [C. Yang, A. S. Dzurak, et al, Operation of a silicon quantum processor unit cell above one kelvin. Nature, 580(7803), 350–354 (2020)], which demonstrates a fully functional unit cell of a Si QDs – based quantum processor operating above 1 Kelvin, an important milestone towards scaling up of Si QDs – based quantum computers.

In this work, two electrons are placed in adjacent potential wells, QD1 and QD2, defined in a silicon-metal-oxide (Si-MOS) device by lithographically fabricated gate electrodes shown in Fig. 1. The electrodes labeled G1 and G2 control the depth of these wells, and the electrode labeled J controls the coupling strength between the wells. A fixed magnetic gradient placed across the wells allows a line carrying a microwave (MW) signal to directly control the spin of one of the qubits via electrically driven spin-resonance (EDSR). The system has both tunable coupling and controls for state preparation. For readout, a single-electron transistor (SET) is placed adjacent to one of the qubits. The SET is a sensitive charge sensor that changes its conductance in response to changes in the quantum dots' charge configuration.

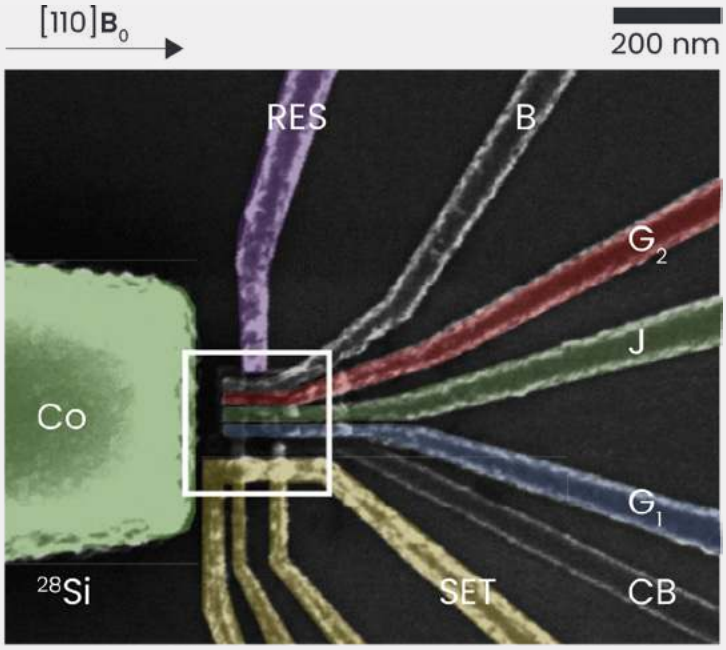


Fig. 1 *A Scanning Electron Microscope image of a two-qubit quantum dot device. The voltages on electrodes G1 and G2 set the detuning €. The RES electrode provides the resonant microwave drive to rotate qubits around the Z-axis. The J electrode controls the coupling between the qubits and is not used in this text. Device readout is done via the Single Electron Transistor (SET) at the bottom of the image. Image supplied by Prof. Andrew Dzurak, UNSW-Sydney.*

STATE PREPARATION, COHERENT CONTROL, AND MEASUREMENT

Many of the operations on the quantum dots involve controlling the **detuning**, ϵ. The detuning is defined as the linear combination of the voltages on G1 and G2 that controls the tilt of the double-well potential towards the (2,0) charge configuration (for positive detuning) or (1,1) charge configuration (for negative detuning). This is usually done at DC, as shown in Fig.2, but can also be modulated as shown below.

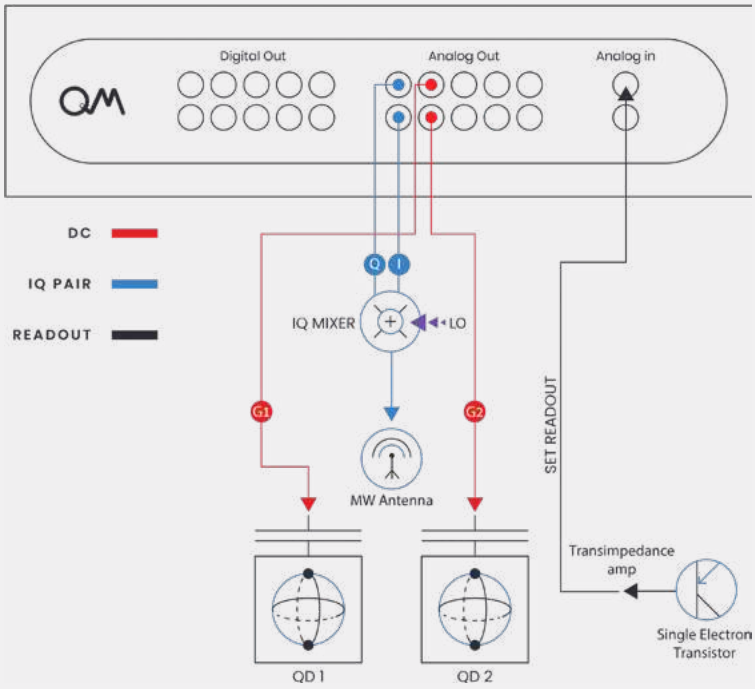


Fig. 2 *A double quantum dot qubit is connected to elements G1 and G2 of the OPX+. The coupling between the dots J is controlled by a third element. The readout of the QD is performed with a single electron transistor (SET). The spin state of each dot can be controlled by a MW signal passed through an antenna.*

The preparation of the system in a well-defined state involves setting the detuning such that both electrons are forced into the same dot, i.e. (2,0) state. After enough time in this charge configuration, the system settles into a singlet spin state. The detuning is then sweeped again to bring the system into the (1,1) state. Thus, the system is prepared in a singlet state with one electron in each dot. Another step, which involves controlling the inter-dot interaction using J, can be used to prepare the system in the state $|\downarrow\downarrow\rangle$. After the preparation, coherent control over each spin can be achieved by driving it at its resonant frequency. Moreover, a CZ gate can be performed by turning on the interaction between the spins via J again.

Finally, the readout is performed by tilting the double-well potential, using the detuning, to a point where the resulting charge configuration is highly dependent on the electrons' relative spin alignment, due to a spin blockade. At this detuning, if the spins are anti-aligned, the electron from QD1 will tunnel to QD2, and the system will be at the (0,2) charge configuration. However, if the spins are aligned, then the electron will not tunnel due to the Pauli exclusion principle and the system will remain in (1,1). Thus, by reading the SET current, we can determine whether the spins are aligned (even parity) or anti-aligned (odd parity).

QUA MACROS

In QUA, repeated sections can be implemented by writing macros. For example, the preparation stage involves placing the two elements at some fixed value and then performing an adiabatic ramp onto the evolution stage. A readout stage can be defined in very similar terms. The code snippet below shows these macros which we reuse in the experiments you'll see in the following sections.

```
def prep_state():
    play('prep', 'G1')
    play('prep', 'G2')
    play('ramp-down-evolve', 'G1')
    play('ramp-up-evolve', 'G2')
def readout_state():
    play('ramp-up-evolve', 'G1')
    play('ramp-down-evolve', 'G2')
    play('readout', 'G1')
    play('readout', 'G2')
```

QUA Code.

The pulses applied to the G1, G2 elements at the preparation, evolution, and readout stages and the difference between them, ϵ, are shown below.

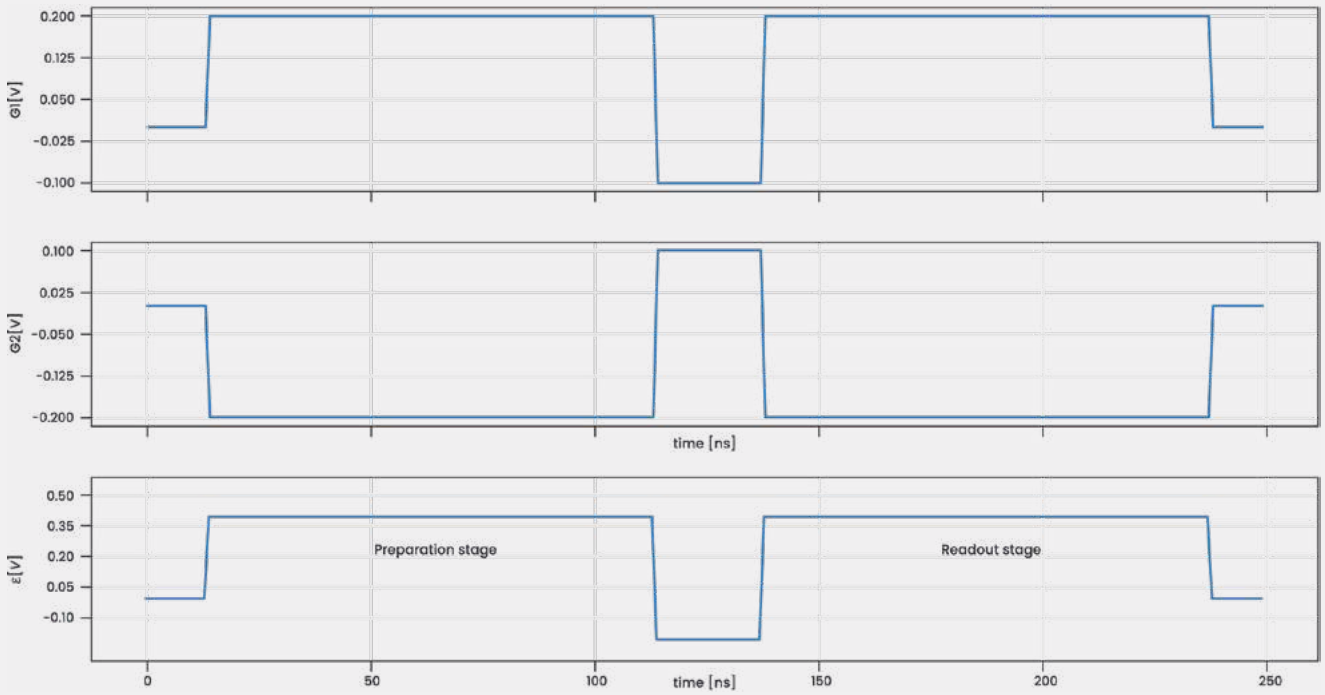


Fig. 3 *Pulse sequence to elements G1 (top) and G2 (middle), and the difference between them giving the detuning (bottom). Setting the voltage on these lines sets the different stages of the experiment, in this case: preparation, free evolution, and readout.*

RABI CHEVRON

One of the most visually appealing basic characterization experiments one can perform on a qubit is generating a Rabi Chevron. Here the system is initialized in a singlet (odd spin parity) state, and the MW drive duration and frequency are scanned. The QUA code below demonstrates how easily we can achieve this with the Quantum Orchestration Platform (QOP). No waveforms need to be precomputed and uploaded since it's now possible to update the duration and frequency of pulses in real-time. We implement this with nested `for` loops, where the outermost loop is used for averaging. The measurement statement is used to integrate the SET current for a duration set by the `measurement_pulse`. Integration is implemented as a weighted sum whose weights are defined by a weights vector `w`.

```
with for_(n,0,n<N_avg,n+1):
    with for_(freq,f_min,freq<f_max,freq+1):
        with for_(d,d_min,d<d_max,d+delta_d):
            update_frequency(freq,'EDS')
            prep_state()
            play('pulse','EDS',duration=d)
            readout_state()
            measure('measure_pulse','SET',integration.full('w',I))
```

QUA Code.

The time saving due to no waveforms being uploaded to the QOP between runs allowed the Rabi chevron measurement in Fig.4, to be taken in **just 4 minutes!**

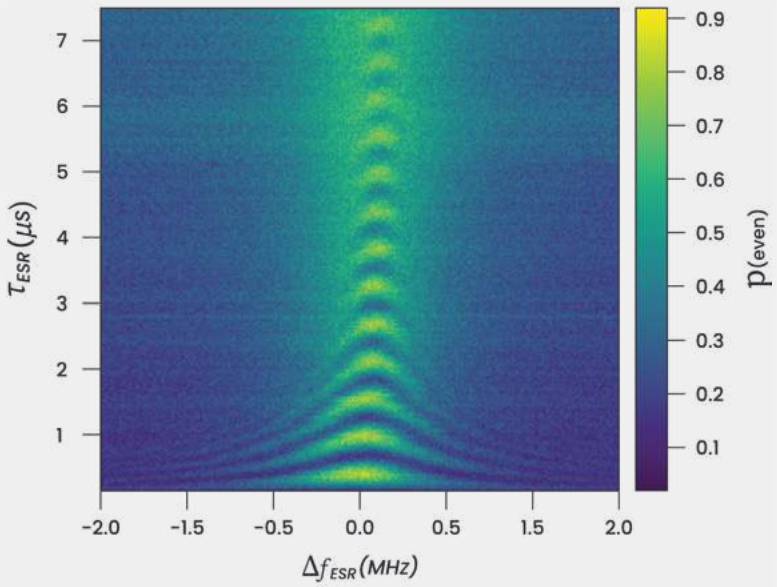


Fig. 4 *Rabi chevron measurement. The probability of an even spin state is plotted against the ESR pulse duration and frequency. This measurement was performed on a grid of 300X400, averaged 100 times, and took approximately 4 minutes to run. Data on Si-MOS quantum dot qubits supplied by Will Gilbert, from the group of Prof. Andrew Dzurak, UNSW-Sydney.*

RAMSEY SPECTROSCOPY AND HAHN ECHO

A more precise way to calibrate the resonant frequency of the Pi pulse is to perform a Ramsey spectroscopy sequence. This essentially generates a "beating" signal between the qubit drive frequency and the qubit's true resonant frequency and can therefore resolve small differences. The Ramsey sequence is composed of an outermost averaging loop, containing an additional `for` loop controlling the delay time between the $\pi/2$ pulses of the sequence. As you can see below, this is very simple to write in QUA.

Each qubit is coupled to a readout resonator and all five resonators are coupled to the same transmission line. The transmission line is probed using another microwave signal that is IQ modulated by two analog output channels of an OPX+ and measured after downconversion by the OPX+ analog input channel.

```
with for_(n,0,n<N_avg,n+1):
    with for_(d,1000,d<10000,d+1000):
        prep_state()
        play('pi_2_pulse','qe')
        wait('qe',d)
        play('pi_2_pulse','qe')
        readout_state()
        measure('measure_pulse','qe',integration.full('w',I))
```

QUA Code.

The trace you see in Fig.5 shows the resulting beating of a QD qubit intentionally detuned from its drive.

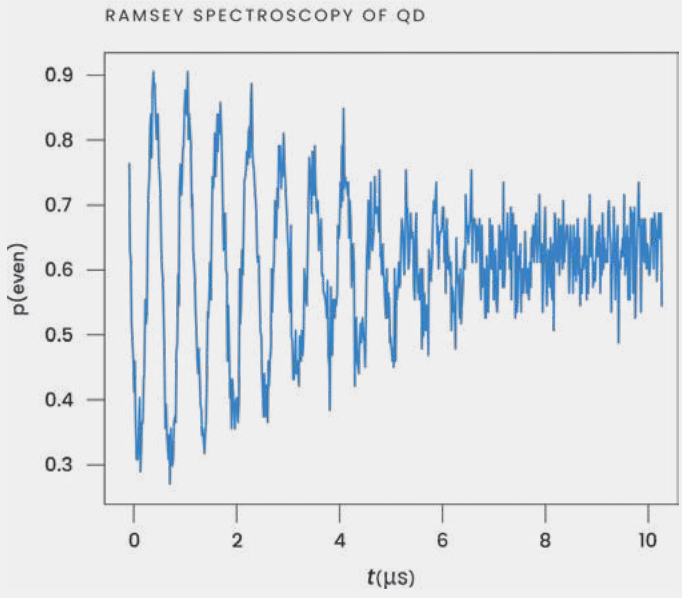


Fig. 5 *Ramsey sequence used to characterize the resonant frequency of a double QD qubit. Data on Si-MOS quantum dot qubits supplied by Will Gilbert, from the group of Prof. Andrew Dzurak, UNSW-Sydney.*

Calibration of a π pulse, obtained by performing the Rabi and Ramsey measurements is crucial for any advanced experimental sequence. Once we achieve this, a natural first step is to measure the rates of decoherence and decay. A Hahn echo sequence, originally used in NMR experiments, is a basic sequence used to characterize the rate of decoherence of a qubit. It can be supplemented by more elaborate pulse sequences such as CPMG or XY-n sequences. The Hahn echo experiment splits the free evolution duration of the Ramsey experiment in two and adds a pi pulse in the middle. This serves to reverse the sense of phase accumulation of the qubit and provide some protection from high-frequency noise components. This reversal indicates an echo in the excited state population after a time 2d (see QUA code snippet below). The decay rate of this echo gives a measure of the low-frequency decoherence rate of the qubit. By modifying the previous code, we can easily describe the experiment in QUA.

```
with for_(n,0,n<N_avg,n+1):
    with for_(d,d_min,d<d_min,d+delta_d):
        prep_state()
        play('pi_2_pulse','qe')
        wait('qe',d)
        play('pi_pulse','qe')
        wait('qe',d)
        play('pi_2_pulse','qe')
        readout_state()
        measure('measure_pulse','qe',integration.full('w',I))
```

QUA Code.

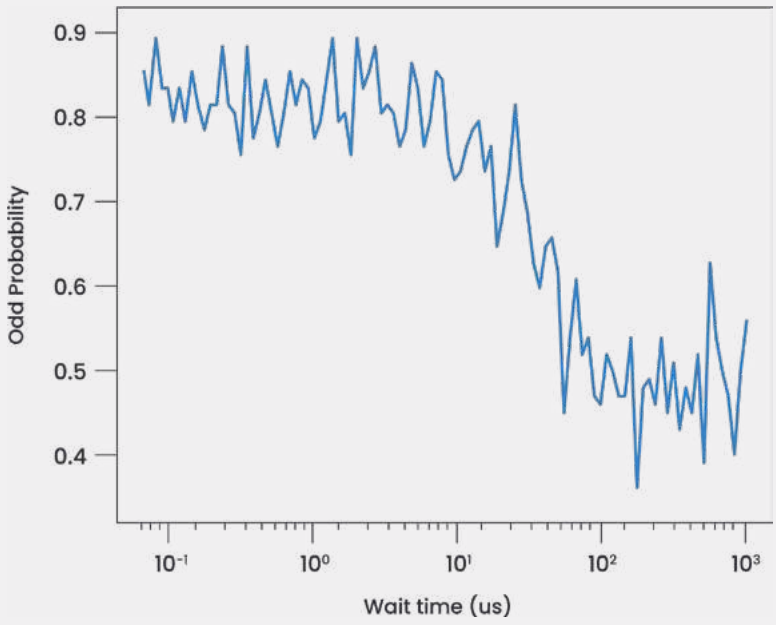


Fig. 6 *Probability of an odd probability state as a function of ESR pulse delay in a Hahn sequence. Data on Si-MOS quantum dot qubits supplied by Will Gilbert, from the group of Prof. Andrew Dzurak, UNSW-Sydney.*

SUPPRESSING QUBIT DEPHASING USING REAL-TIME HAMILTONIAN ESTIMATION

QOP's real-time computational capabilities can be harnessed for a variety of feedback operations. A particularly compelling example of this is employing Bayesian estimation techniques for metrology or, as in this case, estimating the environment of a qubit to extend the coherence time one can achieve.

To demonstrate this, we move to consider another example from [Shulman, M. D. et al, Suppressing qubit dephasing using real-time Hamiltonian estimation, Nature Communications, 5(May), 1–6 (2014)]. In this example, the system comprises an S-T0 qubit residing in a pair of coupled quantum dots in GaAs/AlGaAs. The qubit's state is controlled by the detuning, ϵ, which controls the relative depth of the two potential wells and the coupling between the wells, J.

An external magnetic field is applied (by a superconducting electromagnet) to energetically separate the T+ and T- states from the computational subspace.
In addition, the qubits experience a magnetic field due to the orientation of the nuclear spins in the substrate. By manipulating the qubit state, it is possible to polarize the substrate (so-called "pumping"), thus minimizing the fluctuations, but this has limited effect as there are residual fluctuations. The residual fluctuations are slow in the sense that for each experimental realization the magnetic field will be constant. However, for different realizations its direction is arbitrary and there will therefore be a magnetic gradient across the qubits. This gradient ΔB_z drives a rotation around the x-axis of the S-T0 Bloch sphere shown in Fig.7. Because the magnetic gradient contribution is quasi-static, the system can be operated in sync with the rotation rate it induces. In this work, this is done by using real-time Bayesian estimation to determine the magnetic field gradient, ΔB_z, and thus the associated precession frequency.

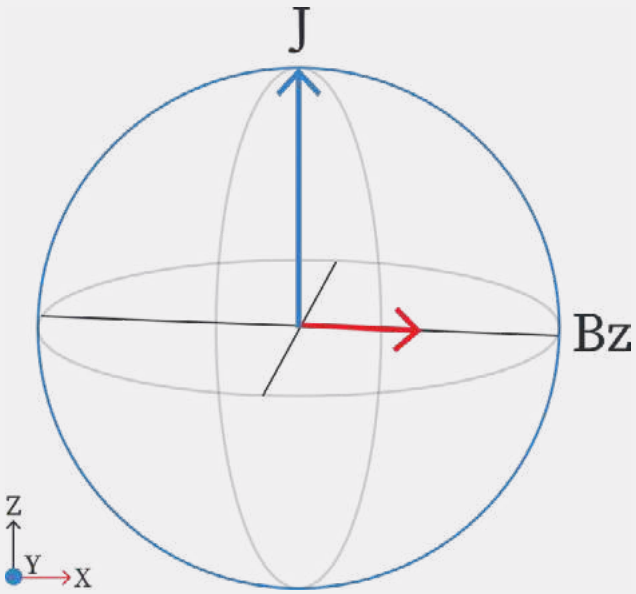


Fig. 7 *Bloch sphere showing the state of two qubits and the driving terms generating rotations. Rotations around the z-axis are driven by the detuning and rotations around the x-axis are driven by the magnetic gradient generated by the nuclear spins.*

Bayesian estimation uses a prior postulate by which all magnetic field values in some range are equally probable and uses Bayes theorem to update the (posterior) distribution repeatedly. The final distribution's most probable field is assumed to be the external field for the experiment's duration.

To perform Bayesian estimation, first, we must perform a single-shot state evaluation. This is done in the first row of the snippet below. The state is measured by measuring the reflection of an RF signal from a Quantum Point Contact (QPC) near the double-dot. The reflected signal is demodulated, and the demodulated value is saved to a variable. We accomplish all this with just a single statement.

The following expression then needs to be repeatedly evaluated (in real-time) for each possible value of the magnetic field:
$P(m_k \mid \Delta Bz) = \frac{1}{2}[1+r_k(\alpha+\beta\cos(2\pi\Delta B_z t_k)]$.

This is done in the first `for` loop. Note the usage of casting and trigonometric functions which are efficiently implemented on the QOP processor. The second `for` loop is a normalization of the resulting probability distribution.

```
1   measure('measure', 'RF-QPC', None, demod.full('integW1', I)
2   assign(state[k - 1], I > 0)
3   save(state[k - 1], satet_str)
4   assign(rk, Cast.to_fixed(state[k - 1]) - 0.5)
5   with for_(fB, fB_min, fB < fB_max, fB + dfB):
6       assign(C, Math.cos2pi(Cast.mul_fixed_by_int(fB, t_samp * k)))
7       assign(Pf[ins1], (0.5 + rk * (alpha + beta * C)) * Pf[ind1])
8       assign(ind1, ind1 +1)
9   assign(norm, 1 / Math.sum(Pf))
10  with for_(ind, 0, ind1 <Pf.length(), ind1 +1):
11      assign(Pf[ins1], Pf[ind1] * norm)
```

QUA Code.

This mathematically clear set of operations is not trivial to implement from scratch on an FPGA which is what was originally demonstrated in this paper. With the QOP, however, no FPGA development skills are required.

References

[1] F. A. Zwanenburg et al., "Silicon quantum electronics," *Rev. Mod. Phys.*, vol. 85, no. 3, pp. 961–1019, Jul. 2013.

[2] C. H. Yang et al., "Operation of a silicon quantum processor unit cell above one kelvin," *Nature*, vol. 580, no. 7803, pp. 350–354, Apr. 2020.

[3] M. D. Shulman et al., "Suppressing qubit dephasing using real-time Hamiltonian estimation," *Nat. Commun.*, vol. 5, no. 1, pp. 1–6, Oct. 2014.

The Quantum Orchestration Platform

AN END TO END QUANTUM CONTROL SOLUTION TO DRIVE THE FASTEST TIME TO RESULTS, AT ANY SCALE

OPX+

RUN STATE OF THE ART EXPERIMENTS WITH EASE

An architecture designed from the ground up for quantum control, the OPX+ lets you run the quantum experiments of your dreams right from the installation. With a quantum feature-rich environment, the OPX+ is built for scale and performance. Now, you can **run the most complex quantum algorithms and experiments in a fraction of the development time**.

PULSE PROCESSING UNIT

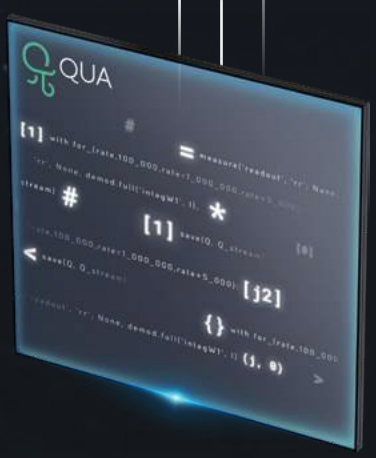
ACHIEVE THE FASTEST TIME TO RESULTS

Within the OPX+ is the Pulse Processing Unit, QM's leading-edge quantum control technology. Progress with incomparable speed and extreme flexibility. Run even the most demanding experiments efficiently, with the fastest runtimes and the lowest latencies in the industry, including quantum protocols that require **real-time waveform generation, real-time waveform acquisition, real-time comprehensive processing, and control flow.**

QUA

CODE QUANTUM PROGRAMS SEAMLESSLY

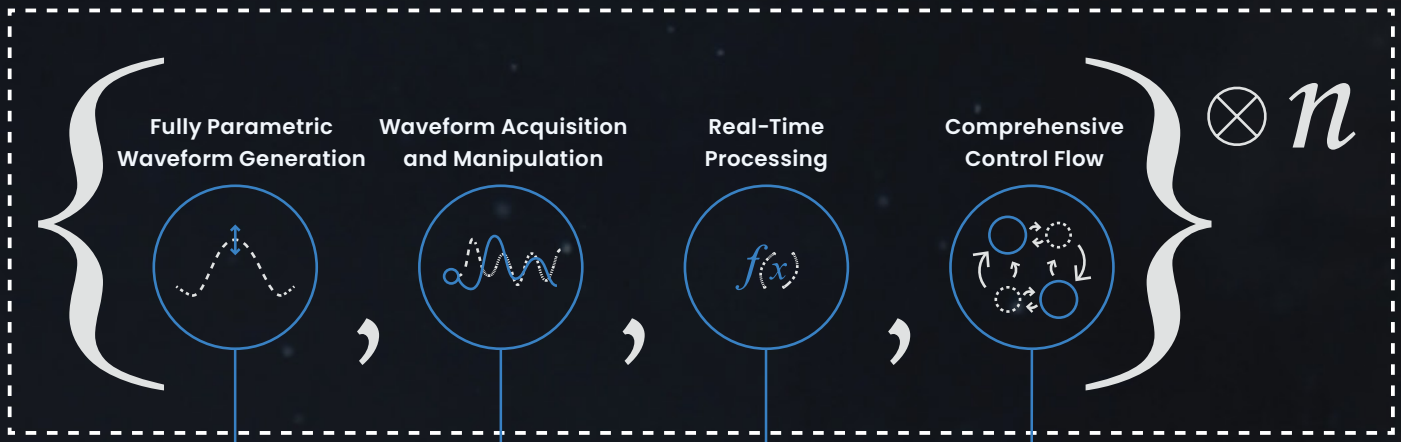
Implement the protocols of your wildest dreams as easily as writing pseudocode. Designed for quantum control, QUA is the first universal quantum pulse-level programming language. Code even the most advanced programs and run them with the best possible performance. Natively describe your most challenging experiments, from complex AI-based multi-qubit calibrations to multi-qubit quantum error correction.



**All of the information above is also valid for the OPX*

QUANTUM MACHINES

YOUR PROTOCOLS LIVE IN THIS PHASE SPACE



Real-Time Waveform Generation

- Fully parametrized: length, frequency, phase (relative & absolute), amplitude, bandwidth, chirp
- Compensations: Crosstalk matrix, FIR and IIR filters
- And many more

Real-Time Measurements

- High fidelity analog to digital conversion
- General integration & demodulation
- Weighted Integrations accumulated integrations, sliced integrations, etc.
- Time Tagging, TTL counting

Real-Time Processing

Turing complete:
- Basic arithmetics
- Evaluation of trigonometric functions
- Vector operations
- Casting of variable types
- And more (Turing complete)

Real-Time Control Flow

- If/else
- For loops
- While loops
- Switch case

Real-Time 'Quantum' Estimations

- State estimations
- Error estimations
- Bayesian estimations
- Correlation functions
- Neural nets based estimations

Real-Time Multi-Qubit Feedback

- Qubit stabilization & tracking
- Quantum error correction
- And many more

THE QUANTUM ORCHESTRATION PLATFORM COVERS THIS SPACE!

- Easily express quantum algorithms and experimental protocols that comprise all of the above.
- Seamlessly sync measurements, real-time calculations, and pulses of different quantum elements.
- Loop over a wide range of parameters in real-time, including intermediate frequencies, amplitudes, phases, delays, integration parameters, measurement axes, etc.
- Use if/else and switch-case statements to condition operations in real time (real time feedback).
- Define procedures (macros) to be reused in the code and access an extensive family of libraries.

QUANTUM MACHINES

If you wish to learn more:

info@quantum-machines.co

About Quantum Machines

Quantum Machines (QM) drives quantum breakthroughs that accelerate the path towards the new age of quantum computing. The company's Quantum Orchestration Platform (QOP) fundamentally redefines the control and operations architecture of quantum processors.

The full-stack hardware and software platform is capable of running even the most complex algorithms right out of the box, including quantum error correction, multi-qubit calibration, and more. Helping achieve the full potential of any quantum processor, the QOP allows for unprecedented advancement and speed-up of quantum technologies as well as the ability to scale into the thousands of qubits. Visit us at: www.quantum-machines.co

---